April 1988
7 DM · 60 öS · 7 sFr

mc

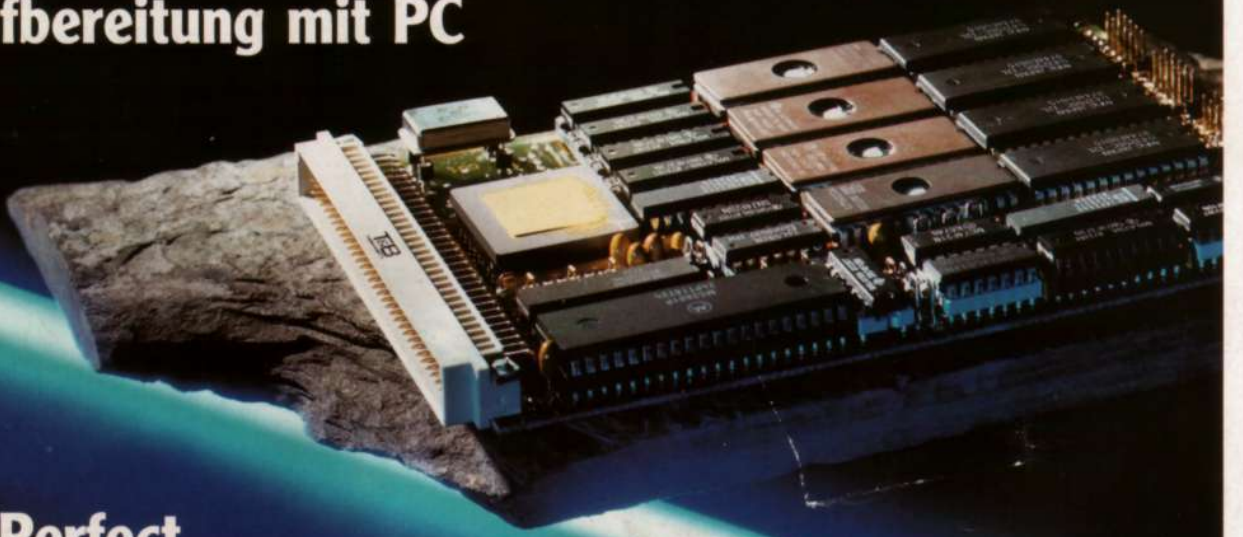
Die Mikrocomputer-Zeitschrift

4/88

Transputern Sie mit:

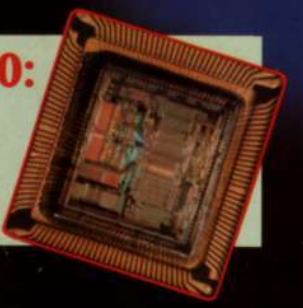
mc-Transputer-Karte

PC-Technik:
Arbeiten mit dem 80287
Bildaufbereitung mit PC



Test:
Word Perfect
Compaq-386 mit VGA

Superprozessor 68030:
Alles über das Design, die logische Struktur und die Leistung



Amiga:
Arbeiten mit der Grafikoberfläche
Intuition in C und Assembler

PLANTRON

Wir machen Erfolg

Alle Computersysteme werden mit ausführlichem deutschen Handbuch geliefert. PLANTRON-Produkte erhalten Sie nur im autorisierten Fachhandel. Bitte fordern Sie Unterlagen zu unserem Gesamtlieferprogramm sowie das Fachhändlerverzeichnis an. Alle Preisangaben sind unverbindliche Preisempfehlungen ohne Monitor.

PT 386 HT mit 128 MB Festplatte und Super-EGA-Karte: DM 9998.–

- 80386 CPU, 16 MHz
- Sockel für 80287 und 80387 Coprozessor
- 1 MB 32-bit-RAM (bis 2 MB on Board)
- Zwei 32-bit-, vier 16-bit- und zwei 8-bit-Slots
- CMOS-Speicher mit Uhr/Kalender
- Parallele und serielle Schnittstelle
- Festplatten-/Diskettencontroller
- 1.2 MB Diskettenlaufwerk
- 128 MB (netto) Festplatte
- Super-EGA-Karte bis 800 × 600 Punkte (GENOA Level 5 kompatibel)
- Große DIN-Tastatur
- Verlagerung des ROM-BIOS und des EGA-ROM in das 32-bit-Extended-RAM (Geschwindigkeitsverbesserung bis 300 % [BIOS] bzw. 1200 % [EGA])
- Erweitertes MS-DOS 3.20 und GW-BASIC
- FTZ-Zulassung

PT-386 HT/2 mit 64 MB Festplatte und Monochrom-Grafikkarte: DM 7998.–

- Sonstige Ausstattung wie PT-386 HT

PLANTRON Computer GmbH, D-6380 Bad Homburg, Telefon: 0 61 72/2 51 88*, Telex: 417 410 placo d, Telefax: 0 61 72/8 10 33

TOP TEAM Rodermark

EDITORIAL

Große Resonanz hatte der erste Teil des Artikels zur Videotext-Schaltung. Viele Leser wollten gleich den zweiten Teil in der Redaktion anfordern. Hier in diesem Heft ist nun der viel begehrte zweite Teil, der die Technik beschreibt. Daß es zu diesem Artikel kam, ist einer Idee von Wolfgang Back, Redakteur beim WDR-Fernsehen zu verdanken. Immer an der besseren Verwertung der Technik interessiert, war es ihm ein Anliegen, das Videotext-Angebot der Rundfunkanstalten nicht im fernseherinternen Videotextbaustein verkommen zu lassen, sondern so nutzbar zu machen, daß zum Beispiel Heimcomputer die Videotext-Seiten abspeichern und weiterbearbeiten können.

Liebe Leser!

Aus seiner Diskussion mit den Autoren des Beitrages (Heinrich Rescher und Eckhard Schadwinkel) entstand die Schaltung, die wir hier präsentieren: Heinrich Rescher hat geschrieben und Erich Schadwinkel hat die Schaltung entwickelt. Das zugehöige Programm für den Mikroprozessor in der Schaltung gibt es gegen einen Unkostenbeitrag in Listingform. Die Bezugsquelle gibt das Sekretariat der mc-Redaktion bekannt.

Wolfgang Back steht immer für guten Technik- und Wissenschaftsjournalismus im Fernsehen. Er hat die Fernsehsendereihe Hobbythek begründet und lange mit Jean Pütz zusammen produziert. Er hat zahllose Einzelsendungen gestaltet und 1983 den WDR-Computer-Club gegründet – eine Sendereihe, die unter den Computerfans große Anerkennung fand. Für all diese (und andere, nicht genannte) Aktivitäten erhielt Wolfgang Back den Journalistenpreis der Zeitschrift Elo, der für die gelungenste Berichterstattung aus dem Gebiet der Elektronik verliehen wird. Das Bild zeigt Wolfgang Back links bei der Preisübergabe anläßlich der Hobbytronic in Dortmund, die Verleger Michael-Alexander Mayer (rechts) und Chefredakteur Burkhard P. Bierschenk vornahmen.

Ein ganz anderes Problem: Viele Privat-Leser haben die Anzeigen von Firmen aus Taiwan in mc gesehen und fragen uns nun, wie sie dort einen PC oder sein Gehäuse kaufen sollen. Zunächst sei gesagt: Wer durchaus will, kann auch eine Privatbestellung mit einer Firma auf Taiwan abwickeln. Aber für eine Einzelbestellung sind diese Anzeigen nicht gedacht. Die Hersteller in Taiwan setzen auf die Bundesrepublik als Wachstumsmarkt in Bezug auf Computer und wissen, daß viele Leser von mc in der Mikrocomputer-Branche tätig sind. Mit den Geschäftsleuten darunter wollen die Firmen aus Taiwan Kontakt bekommen und ihre schönsten Boards in der Bundesrepublik absetzen. Sie selbst können solche Elektronik dann bei einem deutschen Unternehmen kaufen (was sicher auch besser ist, falls einmal eine Reklamation notwendig sein sollte). Zum Beispiel stammen die Module unseres mc-modular-ATs aus Taiwan.

In dieser mc-Ausgabe fängt unsere Transputerserie an. Es gibt sozusagen Transputer zum Anfassen. Wir denken, daß Ihnen der Einstieg auf Maschinensprachebene am nützlichsten ist. Mit einer recht preiswerten Platine, der mc-Transputerkarte, die mit einer RS-232C-Schnittstelle versehen ist. Vom PC und vom Atari aus kann dann das Transputern mit Cross Assembler beginnen. Als Kontrapunkt schildern wir den 68030, den Motorola jetzt im Griff hat, – damit die klassischen Prozessoren zeigen, ob sie mithalten können.

Ihr

Ulrich Rohde

DIE
UNIC-ICB-IDEE

XT/AT-KOMPATIBLE COMPUTER IN INDUSTRIELLEM DESIGN

Die starke Verbreitung von IBM-kompatiblen Rechnersystemen hat es mit sich gebracht, daß eine sehr große Anzahl von entsprechenden Betriebssystemen, Programmiersprachen, Utilities und Programmgeneratoren rasch und preisgünstig erhältlich ist. Die Verwendung dieser Programme für den industriellen Einsatz (Maschinensteuerung, Betriebsdatenerfassung) scheiterte meist daran, daß normale Rechnersysteme für rauhe Umgebungsbedingungen nicht ausgelegt waren (erschütterungsempfindlich, Probleme mit Staub, Temperatur und Feuchtigkeit).

VORTEILE DER UNIC-ICB-RECHNER

100 % softwarekompatibel zu IBM XT/AT-Rechnern. Vollständiger Aufbau auf Doppel- und Einfach-Europakarten – genormte Gehäuse, beliebige Schutzarten. Erweiterter Temperaturbereich durch entsprechende Steckverbindungen und Bauteileauswahl. Erschütterungsunempfindlich durch Einsatz nichtrotierender Massenspeicher (RAM/EPROM-Floppy).
Vollständig modular aufgebautes System, dessen Komponenten für Softwareentwicklung und eventuell Debugging auch am Personalcomputer angeschlossen werden können. Original IBM-Erweiterungskarten (Netzwerkkarten, verschiedene Schnittstellenkarten usw.) sind über eine Adapterkarte direkt auf ICB-Systemen einsetzbar.
Durch die Verwendungsmöglichkeit eines Personalcomputers zur Programmentwicklung für eine industrielle Applikation entfällt die ansonsten notwendige Anschaffung eines eigenen Entwicklungssystems (VME, ECB usw.).
Durch konsequente 19"-Technik (zum Einsatz kommen Platinen im Einfach- und/oder Doppel-Euro-Format) sind ohne großen Aufwand beliebige Schutzarten nach DIN 40 050 realisierbar.

INDUTRONIC®
Dipl.-Ing. Kreiger GmbH & Co. KG
D-8011 Poing/München
Gruberstraße 46
Telefon (0 81 21) 7 01-0
Telefax (0 81 21) 7 13 64

Ich bitte um Zusendung weiterer Informationen:

Name:

Firma:

Straße:

Plz/Ort:

Telefon:

INHALT

INFO

Neues CD-ROM von Sony 12
AT&T steigt bei Sun ein 14
dBase IV kommt im Juli 16
Neues von IBM 22
PS/2-kompatible Systeme 28

GRUNDLAGEN

Die mc-Transputerkarte 38
Völlig neue Hard- und Softwaretechniken müssen mit Transputern eingesetzt werden. mc bringt alles über die RISC-Prozessoren, wie immer leicht verständlich und preiswert nachzubauen

Arbeiten mit dem 80287 45
Mit vielen theoretischen Grundlagen über die Zahlendarstellung in Computern wird der Einsatz des mathematischen Coprozessors beschrieben

Aufbereitung von Bildern 50
Das Digitalisieren von Bildern mit einer entsprechenden Karte an einem PC ist kein Problem mehr. Wir stellen Algorithmen für die Manipulation und die Auswertung der Bilddaten vor

Digitale Bilderzeugung und -verarbeitung 62
Im letzten Teil der Serie geht es um die Software, mit der man den GDP 82786 am mc-68000-Computer betreiben kann

Was Sie über Modula wissen sollten 70
Ausgehend vom Konzept der Unterprogramme wird das Konzept der Module geschildert. Dabei gehen wir auf interne und auf externe Module ein

Der MC68030 76
Die neue Architektur des 32-Bit-Prozessors MC68030 ist so interessant, daß wir dem neuen Prozessor eine dreiteilige Serie widmen

TEST

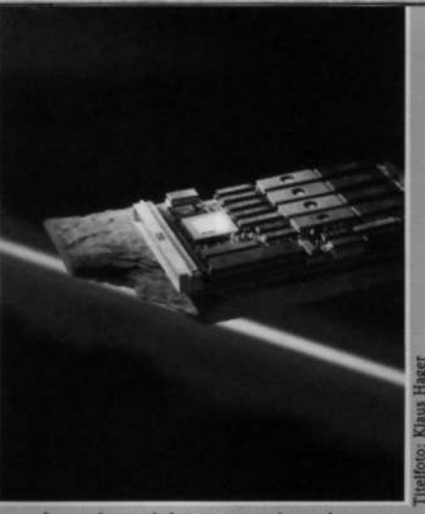
Assembler-Programmierung des Amiga 80
Das Betriebssystem des Amiga ist nicht ohne. Mit zahlreichen Tips wird die Programmierung einfacher

386-PC höchster Leistung 86
Der Compaq 386/20 setzt neue Maßstäbe im PC-Markt. Mit dem Cache-Controller 82385 erreicht das System einen ungewöhnlich hohen Durchsatz. Die neue VGA-Karte macht der Videoausgabe Dampf. Hohe Auflösung und hohe Geschwindigkeit sind beim Compaq 386/20 kein Gegensatz

Der Know-how-Computer im Unterricht 89
Auch mit einer CPU aus Papier kann man rechnen

Kaum kleiner zu kriegen 94
Der wohl kleinste XT-kompatible zeigte im mc-Test, was er kann. Nur wenig mehr als 800 DM kostet das Grundmodell mit einem 3½-Zoll-Laufwerk

C-Compiler Aztec C86 96
Die Programmiersprache C findet auch in der MS-DOS-Welt immer mehr Anhänger. Der C-Compiler von Manx Software Systeme orientiert sich stark an Unix-C-Compilern



Titelfoto: Klaus Hager

Die mc-Transputerkarte

Transputer sind Verwandte der RISC-Maschinen, der 68030 ist eine CISC-Maschine. Erstere besitzen einen Reduced Instruction Set, letztere einen Complex Instruction Set. Wir wollten diese beiden höchstwertigen Mikroprozessoren in dieser Ausgabe nicht gegeneinander ausspielen. Beide sind es wert, daß Sie sich mit deren Architektur auseinandersetzen, um zu wissen, was Mikrocomputer heute leisten können.
Die Riesen-Mikros auf

Seite 38

Aufbereitung von Bildern

Digitalisierte Bilder können durch algorithmische Bearbeitung manipuliert werden. Neben Verfremdungseffekten dient dies in erster Linie der Verbesserung der Bildqualität. Der Kontrast wird erhöht, Details besser erkennbar, Störungen herausgefiltert. Bildreihen veranschaulichen den schrittweisen Einfluß der verschiedenen Verfahren

Seite 50

Der MC68030

Seit einigen Monaten erst wird er in großen Stückzahlen geliefert und schon schickt sich Motorolas jüngster Prozessor an, den Markt von Workstations und VME-Bus-Rechnern bis hinunter zu Atari und Amiga zu durchdringen – kein Wunder, bringt doch dieser Chip ein Maximum an Leistung auf kleinstem Raum zu einem bezahlbaren Preis. Wo genau die Stärken dieses Prozessors liegen, was es mit dem geteilten Cache, der MMU und der Harvard-Architektur auf sich hat, das alles schildert Ihnen ein Insider: Werner Hilf, Schulungsleiter bei Motorola und Autor eines der besten Bücher über den 68000 ab

Seite 76

386-PC höchster Leistung

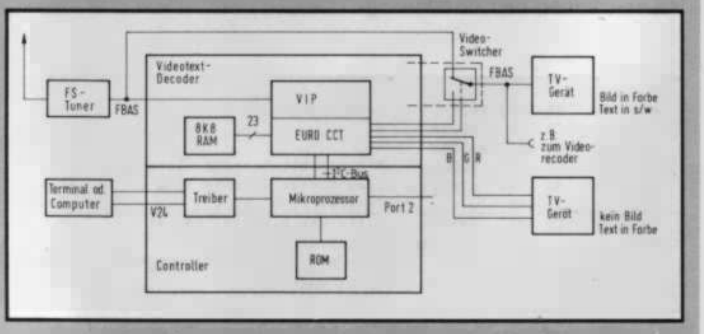
Mit dem Compaq 386/20 stellen wir einen der derzeit leistungsfähigsten Personal Computer vor. Ein Speicher von 32 KByte, der vielen Computern noch vor wenigen Jahren als Hauptspeicher reichen mußte, hat der 386/20 als Cache. Unser Testexemplar wies 5 MByte RAM auf und ließ alle Vergleichs-Computer weit hinter sich.

Seite 86

Tragbare 80286-Power

Nur vom Feinsten haben die Techniker in den Latop Rein 300 SLC eingebaut. Das fängt bei der 20-MByte-Festplatte an und hört beim Plasma-Bildschirm noch lange nicht auf. Wer einen robusten und schnellen Rechner kaufen will, sollte diesen Testbericht lesen.

Seite 104



Videotext in den Mikrocomputer

Mit mc können Sie Videotext besser nutzen. Unser Interface schließen Sie einerseits an die AV-Buchse Ihres Fernsehers an, andererseits an Ihren Computer. Dann können Sie die ausgestrahlten Videotextseiten in den Computer holen, speichern, bearbeiten, ansehen... Wie die Schaltung funktioniert, zeigt der zweite Teil des Artikels. Die Platine gibt es im Handel.

Seite 130

INHALT

Shell mit Editor für Atari ST **98**
Nicht jeder Atari-ST-Anwender ist vom GEM überzeugt. Besonders für Programmierer bietet sich die kommandoorientierte Schnittstelle CRAFT an

Drucker-Sharing mit geringem Aufwand **102**
Ohne zusätzliche Hardware kann man an 14 Rechnern jetzt vier Drucker anschließen und somit besser auslasten

Tragbare 80286-Power **104**
Einen Laptop vom Feinsten präsentieren wir hier. Besonders der Plasmabildschirm hat uns überzeugt

WordPerfect 4.2 **107**
Den – nach Verkaufszahlen betrachtet – Spitzenreiter der Textverarbeitungssoftware haben wir kritisch betrachtet

SOFTWARE

Auch dieser Kreis hat Ecken **106**
Ein weiterer Beitrag zu der unendlichen Geschichte, daß eigentlich jeder Kreis unrund ist

Schnelle Hercules-Routinen **110**
Die Software macht's – auch bei selbstgeschriebenen Programmen. Einige Inline-Routinen für Speed-Freaks

Parameterübergabe bei Turbo-Pascal **112**
Jedes Programm wird durch Assembler-Routinen noch schneller. Wenn man sogar den 8087 ansprechen kann, geht erst so recht die Post ab

Einführung in Intuition **116**
Intuitiv zu bedienen – kompliziert zu programmieren, das ist das Betriebssystem des Amiga. Wir zeigen, worauf man bei der Programmierung achten muß

HARDWARE

mc-EPROM-Brenner für Einchip-Computer **122**
Der mc-EPROM-Brenner zum Anschluß an Standard-PCs mit 4,77 MHz Taktfrequenz ist ein echter Renner! Sogar Einchip-Computer der MCS48-Familie werden durch den mc-EPROM-Brenner programmiert. Wir bringen Grundlagen, Hard- und Software zu diesem Thema

Videotext in den Mikrocomputer **130**
Im zweiten Teil der Serie geht es um den Videotext-Controller und die Software zur einfachen Bedienung

RUBRIKEN

Editorial **3**
Briefe **6, 10**
Bücher **37**
Info **12**
Spruch des Monats **95**
Zitat des Monats **100**
Markt **144**
Vorschau **182**
Impressum **6**
Verlagsinformationen **181**

mc
Die Mikrocomputer-Zeitschrift

Franzis-Verlag GmbH
Karlstraße 37–41, 8000 München 2
Postfach 37 01 20, 8000 München 37
Telefon (0 89) 51 17-1
Telex 5 22 301
Telefax (0 89) 51 17-3 79
Tedas (0 89) 59 84 23 und 59 64 22

Chefredakteur:
Dipl.-Math. Ulrich Rohde

Stellvertretender Chefredakteur:
Dipl.-Ing. (FH) Ulrich Kruppe

Redaktion:
Redaktionsassistentin:
Rita Schleser (0 89) 51 17-3 54
Ressortredakteure: Horst Brand, Dipl.-Ing. (FH) Wolfgang Hascher (fl), Aktuelles: Reiner Schönrock, PC-Technik: Dipl.-Ing. (FH) Dieter Strauß
Anschrift der Verantwortlichen siehe Verlagsanschrift

Korrespondent: USA:
Ray Duncan

Ständiger Mitarbeiter:
Dipl.-Ing. (FH) Herwig Feichtinger (zu erreichen unter der Anschrift der Redaktion)

Art Direktion:
Dipl.-Designer (FH)
Volker Hilbel

Layout, Grafik, Herstellung:
Josef Wurzinger

Franzis-Labor:
Neucom electronic GmbH

Software Service:
SHAMROCK Software Vertrieb
(0 89) 59 54 68

Verantwortlich für den Inhalt:
Dipl.-Math. Ulrich Rohde

Verantwortlich für Anzeigen:
Dietger Kötter

Gesamtherstellung:
Franzis-Druck GmbH
Karlstraße 35
8000 München 2
Telefon (0 89) 51 17-1

© 1988 für alle Beiträge bei Franzis-Verlag GmbH

ISSN: 0720-4442

Vertriebskennzeichen:
B 7745 E

Sonderdrucke:
Jakob Wintersberger

Lizenzen:
Georg Geschke, Königstr. 8,
3000 Hannover 1,
Telefon (05 11) 34 53 62

Urheberrechte:
Die in mc veröffentlichten Beiträge sind urheberrechtlich geschützt. Alle Rechte, insbesondere das der Übersetzung in fremde Sprachen, vorbehalten. Auch die Rechte der Wiedergabe durch Vortrag, Funk- oder Fernsehsendung im Magnettonverfahren oder ähnlichem Wege bleiben vorbehalten. Fotokopien für den persönlichen oder sonstigen eigenen Gebrauch dürfen nur von einzelnen Beiträgen oder Teilen daraus als Einzelkopien hergestellt werden.

BRIEFE

C-Compiler im Vergleich

In diesem Artikel (mc 3/88) ist eine falsche Information enthalten: Es trifft nicht zu, daß die Firma Philgerma das deutsche Handbuch für den Lattice C Compiler (Ataria ST) auch einzeln für 28 DM verkauft. Dieses Handbuch ist die offizielle Übersetzung des englischen Handbuchs und wird von zwei Händlern, darunter Philgerma, nur zusammen mit dem Lattice C Compiler vertrieben.
Bitte informieren Sie ihre Leser entsprechend.

Helge Raab,
Philgerma,
8000 München

Im Zeichen von OS/2

In Ihrem Bericht von der Comdex '87 (mc 1/88) schreiben Sie: „Neben Atari und Commodore bringen jetzt auch Unisys, Grid, Tandon, Compaq, AST, Samsung und Kaypro, um nur einige zu nennen, neue Systeme mit diesem Prozessor (dem 80386) auf den Markt. Welche dieser Computer dann tatsächlich in Deutschland zu haben sein werden, wird frühestens die CeBIT im Frühjahr in Hannover zeigen. Vorher dürfte sowieso kaum ein Hersteller seine Produkte marktreif bekommen."
Es ist zweifelsfrei richtig, daß jetzt diverse Hersteller Personal Computer mit dem 80386-Prozessor auf den Markt bringen. Allerdings hat Compaq bereits im September 1986 als erster namhafter Hersteller einen PC mit dem 80386-Prozessor in der 16-MHz-Version auf den Markt gebracht, nämlich den Compaq Deskpro 386. Systeme mit der 20-MHz-Version des 80386, den Compaq Deskpro 386/20 und den Compaq Portable 386 liefert Compaq bereits seit September 1987 weltweit aus, selbstverständlich auch in Deutschland. Im übrigen stellt Compaq grundsätzlich nur dann neue Produkte vor, wenn diese marktreif sind und auch in den vom Markt verlangten Stückzahlen geliefert werden können. Nach Angaben des unabhängigen amerikanischen Marktforschungsinstituts Store-Board hatte Compaq im vergangenen Jahr mit den Produkten Compaq Deskpro 386, Compaq Deskpro 386/20 und Compaq Portable 386 einen Marktanteil von 80 Prozent bei über den Händlerkanal verkauften PCs mit dem 80386-Prozessor.
Nur der Vollständigkeit halber möchten wir anmerken, daß auch unter der Betriebssystemumgebung MS Windows 386 mehrere MS-DOS-Programme gleichzeitig ablaufen können.

Edmund Hain,
COMPAQ Computer GmbH,
8000 München

Programmierstil

Auf die Gefahr hin, eine Diskussion in die Länge zu ziehen, möchte ich einen Beitrag zum Thema „Programmierstil" liefern: Während meiner Erkundungsreisen durch das Betriebssystem des HX-20 bin ich auf die Routine in *Bild 1* gestoßen. Es scheint eine einfache Routine zu sein, die das nächste Zeichen daraufhin untersucht, ob es „)" ist – wenn da nicht die beiden Zeilen mit dem „*" im Kommentar wären. Diese Zeilen erscheinen hier überflüssig, aber auch nicht störend. Ihr Sinn klärt sich auf, wenn man ab Adresse 88D0 disassembliert *(Bild 2)*: Die gleiche Routine überprüft nun nicht mehr auf „)" sondern auf „(". Analog dazu prüft die Routine beim Einsprung auf Adresse 88D3 auf „,".
Ein Trick, der nach außen eine Struktur der Routine vorspiegelt:
$88CD Prüft ob „)"
$88DO Prüft ob „("
$88D3 Prüft ob „,"
– aber intern genial (unstrukturiert) realisiert.

Gerd Güldenpfennig,
7500 Karlsruhe

Anmerkung der Red.: Diese Art der Programmierung ist keine Spezialität des HX-20, ähnliche Konstruktionen finden sich z. B. in den Microsoft-Basic-Interpretern für den 6502 gleich dutzendweise. Doch in einer Zeit, in der 64-KByte-EPROMs kaum teurer sind als 16-KByte-Typen (billiger als ein 2708 je war!) und die Adreßräume der Prozessoren in ganz anderen Größenordnungen liegen, ist ein solches „Byte-Knautschen" nicht nur überflüssig, sondern in den meisten Fällen sogar unwirtschaftlich. Dabei mag man durchaus bedauern, daß solche kleinen Kunstwerke des Programmierens nicht mehr gefragt sind – ändern wird man es wohl kaum.

```
88CD C6   29 LDAB #$29       Lade B mit $29 (29=")")
88CF 8C C628 CPX  #$C628     *
88D2 8C C62C CPX  #$C62C     *
88D5 DE   FB LDX  $00FB      $FB enthält Adresse nächstes Zeichen
88D7 A6   00 LDAA $00,X      Lade dieses Zeichen in A
88D9 11      CBA             Vergleiche A und B
88DA 26   03 BNE  $88DF      Falls ungleich ==) Fehler
88DC 39      RTS  (Weiter)   Programmfortsetzung
88DF C6   02 LDAB #$02       Der Fehlercode "02"(Syntax-Error) wird
88E1 7E 8433 JMP  $8433      hier ausgegeben
```

Bild 1. Was auf den ersten Blick wie ein Irrtum des Programmierers anmutet ...

```
88D0 C6   28 LDAB #$28       Lade B mit $28 (28="(")
88D2 8C C62C CPX  #$C62C     *
88D5 DE   FB LDX  $00FB
88D7 A6   00 LDAA $00,X
88D9 11      CBA
88DA 26   03 BNE  $88DF
88DC 39      RTS  (Weiter)
88DF C6   02 LDAB #$02
88E1 7E 8433 JMP  $8433
```

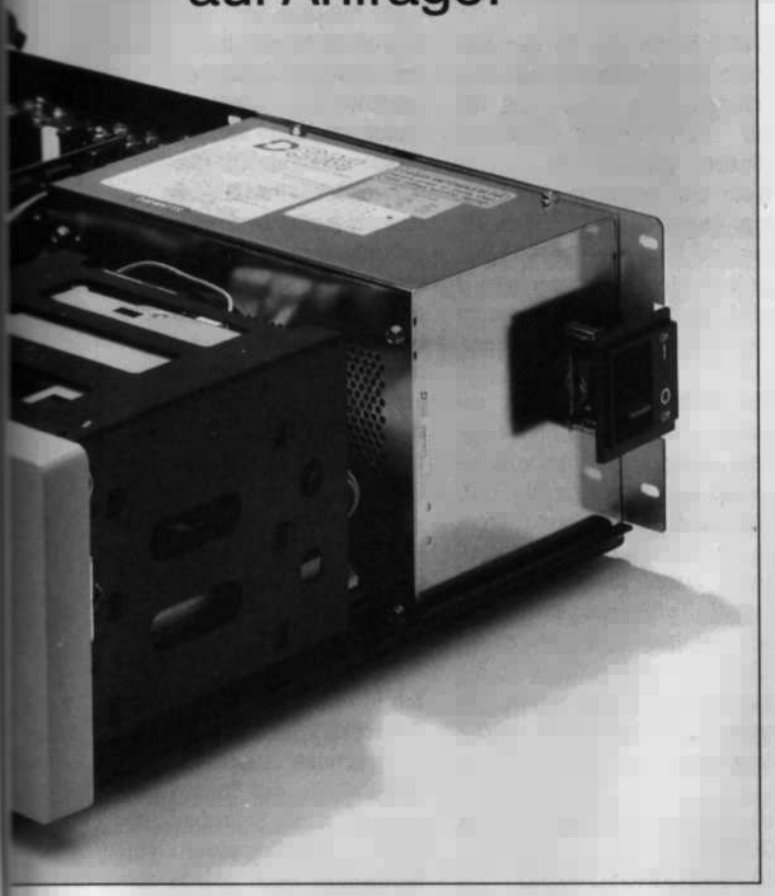
Bild 2. ... entpuppt sich als versteckter Einsprung in die gleiche Routine

m Modular-Konzept

6

NEU!

OEM-Konditionen
auf Anfrage!



Fordern Sie unsere ausführlichen Listen (PC-News und PC-Software-News mit über 500 Programmen) an!

Händleranfragen erwünscht. (Bitte mit Händlernachweis)

DSM Digital Service GmbH

Zentrale: Landwehrstraße 37 · 8000 München 2
Filiale: Schillerstraße 17 · 8000 München 2

DSM **Tel. 089/596916**
Telex 523545 dsm d · Fax 089/598261

Computer Systeme

Ein Unternehmensbereich der *DSM Digital* Service GmbH

Bei den hier aufgeführten Karten handelt es sich um qualitativ hochwertige Module, die auch in herkömmlichen IBM- oder IBM-kompatiblen Systemen einsetzbar sind. Auf alle Komponenten ½ Garantie! Alle Preise inclusive 14 % MwSt.
IBM ist eingetragenes Warenzeichen der IBM Corporation.

1 DSM AT286-modular-Fertiggerät bestehend aus AT-modular-Board V 2.0, 512 KB RAM (100 ns), BUS-Platine, Festplatten-Floppy-Controller, Monochrome-Graphik-Karte, stabiles Gehäuse, 200 Watt Netzteil, MF-2-kompatible Tastatur (102 Tasten), 5¼" 1.2 MB Floppy-Disk-Laufwerk, 20 MB Seagate-Festplatte (65 ms), 14"-Monitor. Mit MS-DOS 3.3®, GW-Basic, und deutschen Handbüchern. Komplett aufgebaut und geprüft (48 Std. Burn-In), 1 Jahr Garantie
Best.-Nr. M0203 **DM 4.198,–**

2 DSM AT286-modular-Kit 2
Wie Pos. 1 als Teile-Kit
Best.-Nr. M0202 **DM 3.798,–**

3 DSM AT286-modular-Kit 1
Wie Pos. 2 ohne Hard-Disk, ohne MS-DOS und Monitor
Best.-Nr. M0201 **DM 2.798,–**

4 AT286-modular-Board V2.0
auf 1 MB RAM ausbaubar, mit 512 KB Speicherbausteinen bestückt, 7 Kanal DMA, 16 Interrupt-Ebenen, 3 programmierbare Timer, 1 Realtime-Clock, CMOS-RAM, bis 64 KB ROM, Keyboard-Controller, 10 MHz, 0-Wait! Neuer CHIPS-TECHNOLOGY-Satz und mit Standard-Speicherbausteinen bestückbar. Award-BIOS. Inkl. Handbuch
Best.-Nr. M0102, 100ns RAMS's **DM 1.298,–**
Best.-Nr. M0100, 120ns RAM's **DM 1.198,–**
Best.-Nr. M0101, ohne RAM's **DM 898,–**

5 AT386-modular-Board
Hochleistungs-CPU-Karte mit 80386 Prozessor (32-Bit) 16 MHz. Bis zu 16 Mbyte-0-Wait-Speicher, Sockel für 80287 und 80387 Coprozessor
Best.-Nr. M0142 **DM 3.798,–**

6 Bus-Platine
mit insgesamt 8 Steckplätzen, davon 6 AT-Slots sowie 2 XT-Slots
Best.-Nr. M0103 **DM 98,–**

7 Festplatten-Floppy-Controller
Western Digital®-Chip-Satz. Verwaltet bis zu 4 Laufwerke, 2 Floppy-Laufwerke (360 KB/1.2 MB) und 2 Hard-Disk-Laufwerke. Optional 1 zusätzliches Floppy-Laufwerk
Best.-Nr. M0129 **DM 398,–**

8 Monochrom-Grafik-Karte
HERCULES™-kompatibel, mit parallelem Druckeranschluß
Best.-Nr. M0107 **DM 148,–**

9 EGA-Standard-Grafik-Karte
EGA 640x350, 640x200, 320x200, CGA, MDA, 16/64 Farben
Best.-Nr. M0108 **DM 398,–**

10 SUPER-EGA Level 5
800x600, 640x480, 640x350, 640x200, 320x200, 16/64 Farben
Best.-Nr. M0111 **DM 598,–**

11 RAM-Karte
2 MegaByte Extended und Expanded Memory, 10 MHz, (0 KB-RAM) AT-RAM-Erweiterungs-Platine
Best.-Nr. M0112 **DM 398,–**

12 Gehäuse
Stabile Ausführung mit Befestigungsmaterial
Best.-Nr. M0122 **DM 198,–**

13 Netzteil
200 W, kurzschlußfest incl. Netzkabel
Best.-Nr. M0128 **DM 238,–**

14 Tastatur
MF-2-kompatibel, 102 Tasten
Best.-Nr. M0126 Deutsch **DM 298,–**
Best.-Nr. M0127 ASCII **DM 298,–**

15 3½" 1.44 MB Floppy-Disk-Laufwerk
OS/2®-Format (2 MB unform.)
Best.-Nr. M0104 **DM 398,–**

16 3½" 1.2 MB Floppy-Disk-Laufwerk
AT-kompatibel (1.6 MB unform.)
Hersteller: TEAC/NEC
Best.-Nr. M0105 **DM 298,–**

17 5¼" 1.2 MB Floppy-Disk-Laufwerk
AT-kompatibel (1.6 MB unform.)
Hersteller: NEC
Best.-Nr. M0106 **DM 298,–**

18 Festplatten-Laufwerke
a) 20 MB Seagate (65 ms)
Best.-Nr. M0123 **DM 598,–**
b) 40 MB Seagate
Best.-Nr. M0124 **DM 1.150,–**
c) 80 MB Seagate (28 ms)
Best.-Nr. M0125 **DM 2.498,–**

19 Coprozessoren
80287-1 (10 MHz)
Best.-Nr. M0118 **DM 698,–**
80387 (16 MHz)
Best.-Nr. M0119 **DM 1.398,–**

20 Monitore
14" Monitore mit Schwenkfuß
a) grün
Best.-Nr. M0134 **DM 325,–**
b) bernstein
Best.-Nr. M0135 **DM 329,–**
c) **EGA-Farb-Monitor**
Auflösung bis 640x350 Punkte
Best.-Nr. M0136 **DM 998,–**
d) **EIZO-Farb-Monitor**
Auflösung bis 820x620 Punkte für den Betrieb aller EGA-Karten, TTL-Eingang, umschaltbar auf ANALOG-Eingang
Best.-Nr. M0137 **DM 1.798,–**

21 Betriebssystem
MS-DOS 3.3® deutsch mit GW-BASIC und deutschen Handbüchern
Best.-Nr. M0120 **DM 298,–**

22 Zubehör
a) Handbuch AT-modular-Computer im praktischen Kunststoffordner
Best.-Nr. M0130 **DM 39,–**
b) Kabelsatz (2x Floppy und 1x Hard-Disk)
Best.-Nr. M0131 **DM 49,–**

mc-68000-Computer

Von der Fortschrittlichkeit und der Qualität des mc-68000-Computers überzeugt, habe ich mir dieses System vor einigen Jahren auf dem Gebrauchtmarkt zugelegt. Eigentlich finde ich auch noch heute das Konzept des Systems gut, fühle mich jetzt jedoch etwas alleine gelassen. Zwar ist eine pauschale Verurteilung nicht angebracht: Es gibt auf dem Markt eine Menge Software, die ich einmal als „Grundsoftware" bezeichnen möchte. Hierzu zähle ich Betriebssoftware ebenso wie die zur Verfügung stehenden Compiler in den einzelnen Betriebssystemen (OS9/68k und CP/M-68k). Jedoch vermisse ich Angebote von fertiger Anwendersoftware, die auch von der Preisgestaltung zum Hobbymarkt paßt. Man kann einem Hobbyisten nur bedingt zumuten, daß er sich selbst aufwendige Kalkulationsprogramme in einer höheren Programmiersprache erstellt. Auch sind nicht alle Besitzer dieses Systems in der Lage, andere Softwarepakete, von denen das Listing vorhanden ist, auf den mc-68000-Computer anzupassen.
Geht es Ihnen, lieber Leser, ähnlich? Dann schreiben Sie dies doch. Ich bin an einem regen Erfahrungsaustausch interessiert, um mein mc-68000-System nicht in der Ecke veröden lassen zu müssen. Ein Austausch ist sicherlich über die Redaktion der mc möglich, jedoch können Sie mich auch gerne direkt anschreiben.

Reiner Lins,
Rich.-Wagner-Str. 7,
6053 Obertshausen

Turbo-Basic-Erfahrungen

Nachdem ich etliche positive Testberichte über Borlands Turbo-Basic gelesen habe, stellen sich mir einige Fragen. Eine deutsche Version 1.0, die ich vor einiger Zeit mit bewährten GWBASIC-Programmen testen konnte, im Directory datiert mit dem 21. 4. 87, zeigte folgende Merkwürdigkeiten:
Bei der zeichenweise erfolgenden Konvertierung einer 72 KByte langen Wordstar-Datei brach eine WHILE-WEND-Schleife, die ihrerseits in eine FOR-NEXT-Schleife eingebettet war, nach genau 65 536 Durchläufen ab, was vermutlich auf den Integer-Bereich des Compilers zurückzuführen ist; die Schleife enthielt aber keine Zählvariable. Microsofts BASCOM führte dasselbe Programm fehlerlos in weniger als der Hälfte der Zeit aus.
Bei der Ausgabe eines String-Feldes auf dem Bildschirm mit PRINT wurden unerklärlicherweise vor allen Strings mit mehr als 80 Zeichen Länge Leerzeilen eingefügt, die weder unter GWBASIC noch unter BASCOM auftraten. Dieses Problem war trotz mehrerer Versuche nur durch eine zusätzliche Abfrage der String-Länge vor der Ausgabe und ein bedingtes Zurücksetzen des Cursors um eine Zeile zu lösen, was natürlich eine Verlangsamung bewirkte. Der Hex-Code der Strings zeigte keine Besonderheiten. Da die Strings ansonsten korrekt behandelt wurden, dürfte die Ausgabefunktion fehlerhaft sein.
Beim Wechsel zwischen EDIT und RUN kam es trotz NEW zu Bildschirm-Ausgaben, die aus der gelöschten Programmversion stammten und mit der aktuellen Version nichts mehr zu tun hatten. Hier dürften Pointer falsch initialisiert sein.
Der Versuch, den SHELL-Befehl in Programmen oder auf der Editor-Ebene zu verwenden, scheiterte mit DOS 2.11, weil Turbo-Basic hier wohl den Interrupt 4Bh aufruft, der erst ab DOS 3.X richtig funktioniert. Unter GWBASIC funktionierte der SHELL-Befehl mit allen mir zugänglichen DOS-Versionen. Dieses Problem dürfte Benutzer älterer DOS-Versionen auch bei der Arbeit mit Turbo-Pascal 4.0 und Turbo-C treffen. Vor allem auf Laptops mit speziell angepaßtem DOS 2.11 ist der Ärger vorprogrammiert. Dies kann man allerdings nicht Borland anlasten.
Weiterhin ist es für Interessenten wichtig zu wissen, daß der Compiler allein schon weit über 200 KByte Speicher belegt. Was die Fehler betrifft, frage ich mich, ob ich hier zufällig auf eine fälschlicherweise nicht zurückgezogene Vorversion gestoßen bin.

Harald Lukas-Simonyi,
Mönchengladbach

Die Antwort von Heimsoeth

Leider liegen uns zu den von Ihnen erwähnten Programmen keine Listings vor, so daß wir nur Vermutungen anstellen können.
Wenn ein Programm eine Datei zum Lesen oder Schreiben öffnet und Daten in den I/O-Puffer liest, ohne daß dieser geleert wird, so bricht das Programm ab, sobald dieser Puffer überläuft.
Die Problematik entsteht dadurch, daß Basic-Interpreter zwar einen I/O-Puffer von bis zu 255 Byte benutzen (WIDTH-Statement), aber nicht prüfen, ob zu lesende oder zu schreibende Daten in diesen Puffer passen. Turbo-Basic besitzt einen Dateipuffer von bis zu 64 KByte und prüft explizit, ob die Daten in den Puffer passen. Ein typisches Beispiel für solche Programme:

```
10 OPEN „TEST.DAT" FOR
OUTPUT AS #1
20 B$=STRING$(100,32)
30 FOR I%= 1 TO 7000
40 PRINT #1, B$;
50 NEXT I%
60 CLOSE
70 END
```

Dieses Programm bricht ab, da der Puffer nur voll geschrieben wird, ohne daß der Puffer jemals entleert wird. Um den Puffer zu leeren, muß ein CR (Carriage Return) folgen. Das Programm wird sofort lauffähig, wenn man den Strichpunkt am Ende der PRINT #1 Anweisung wegläßt, denn dadurch wird nach jeder PRINT #1 Anweisung ein CR ausgeführt und der Puffer dadurch geleert.

Bezüglich der Geschwindigkeit der Programme würde ich annehmen, daß das Programm etwas „schludrig" programmiert ist und die Schleifenvariable nicht als Integervariable deklariert ist, was bei einem Compiler natürlich das Laufzeitverhalten ändert.
Der beschriebene Fehler bei der Ausgabe von Strings trat nur in der ersten Version von Turbo-Basic auf und ist mittlerweile behoben.
Das RUN-Fenster im integrierten Entwicklungssystem wird explizit nur dann gelöscht, wenn in den verschiedenen Programmen ein CLS-Statement verwendet wird. Dies ist nach unserer Auffassung sehr vorteilhaft, da man somit Ergebnisse eines alten Programmes selbst noch dann zu Prüfzwecken sehen kann, wenn man inzwischen ein neues Programm bearbeitet. Wird dies nicht gewünscht, so muß am Anfang des neuen Programmes nur ein CLS stehen.

H. Paintner,
Heimsoeth software, München

Wer kann helfen?

Ich bin seit kurzem Leser Ihrer Zeitschrift. Aus den mir bis jetzt vorliegenden Ausgaben ersehe ich, daß Sie des öfteren Anleitungen zum Selbstbau bzw. Umbau von Computern veröffentlichen. Mich würde daher interessieren, ob Ihnen Schaltpläne für den Speicherplatzausbau eines Apple Macintosh 512k RAM, 64k ROM zur Verfügung stehen, oder ob Sie eine Quelle wissen, bei der ich solche Pläne erwerben könnte. Da ich in Kürze eine SCSI-Festplatte anschließen möchte, ist es notwendig, den Rechner auf 1MB RAM, 128K ROM aufzurüsten.

Martin Zang,
8069 Scheyern

Basic-Singlechip

Einplatinencomputer mit dem 8052 AH-Basic selbstgebaut und programmiert. Von Rudolf Busch und Heiko Requardt. 194 Seiten, 72 Abbildungen, 12 Tabellen, kartoniert, 48 DM, Franzis-Verlag, München, 1987.
ISBN 3-7723-8741-1

Daß der Aufbau, die Programmierung und die Installation von festprogrammierten Steuerungscomputern keine schwierige Aufgabe ist, wissen unsere Leser spätestens seit der Vorstellung des mc-Basic-EMUF (mc 12/86). Auf der Basis des Prozessors 8052 AH-Basic bauen die Autoren ein universelles System auf und erklären dabei alles, was man über diesen Chip und seinen Einsatz wissen muß. Von einer Kalenderuhr über die A/D-Wandlung bis zur Steuerung von Maschinen reicht das Spektrum der vorgestellten Applikationen. Neben den Schaltplänen sind Platinenlayouts und Programm-Listings enthalten.
Wer mit wenig Aufwand zu einer flexiblen Computer-Steuerung kommen möchte, muß das Buch „Basic-Singlechip" lesen.

Rs

Singlechip-Prozessoren

On-Chip-Peripherie und Kommunikationsmöglichkeiten moderner Mikrocontroller. Von Erich Esders. 142 Seiten, 67 Abbildungen, kartoniert, 48 DM, Franzis-Verlag, München, 1987.
ISBN 3-7723-8611-3

Der Einsatz von Singlechip-Prozessoren läßt sich wesentlich in zwei Gruppen gliedern: Zum einen finden wir sie in komplexen Computern, wo sie deutlich umrissene Aufgaben übernehmen und somit die CPU entlasten. Das andere Einsatzgebiet liegt bei Steuerungsapplikationen, wo sie eigenständig Abläufe überwachen.
Nach einer allgemeinen Vorstellung von Singlechips und deren Peripherie wird am Beispiel der 6805- und der 8051-Familie das Thema konkretisiert. Erich Esders geht sehr detailliert auf die Programmierung der Prozessoren ein und zeigt, wie man Programme entwickelt und testet.
Der Autor hat ein Handbuch für alle Entwickler geschaffen, die nicht täglich mit Singlechip-Prozessoren zu tun haben. Die Ausführungen sind industrieorientiert und nicht an spezifische Typen gebunden. *Rs*

Grundlagen der PC-Handhabung

Eine strukturierte Einführung in das Standard-Betriebssystem MS-DOS. Peter Freese, Rowohlt Taschenbuch Verlag, Hamburg, 1987, 254 Seiten, 14,80 DM.
ISBN 3-499-181-452

Das Buch von Peter Freese ist für alle diejenigen gedacht, die eine erste und verständliche Anleitung zum Arbeiten mit dem PC und dem Betriebssystem MS-DOS brauchen, ohne gleich im Fachchinesisch baden zu gehen. In Form eines Grundkurses wird in gutem Deutsch Wissen aus der Computerpraxis vermittelt. Anhand von Abbildungen und praktischen Beispielen wird zum Ausprobieren angeregt. Begriffe, die zum Alltag bei der Arbeit am PC gehören, werden nicht einfach vorausgesetzt, sondern eingeführt und erläutert. Durch Merksätze und Zusammenfassungen wird Wesentliches gefestigt. Diese Zusammenfassungen sind durch Fettschrift hervorgehoben oder durch Balken markiert. Die zu lernenden Befehle sind grau hinterlegt. Dadurch werden Kernsätze beim Nachschlagen leichter gefunden. Darüber hinaus gibt es eine Kurzübersicht über wichtige MS-DOS-Befehle im Anhang des Buches. Sollten die Kurzbeschreibung der Wirkung und ein praktisches Beispiel nicht ausreichen, hilft die angefügte Seitenangabe beim gezielten Nachschlagen im Buch. In dieser Kurzübersicht finden sich auch Befehle, die im Buch nicht behandelt werden. Dem Leser soll sicherlich dadurch klar gemacht werden, daß es sich bei diesem Buch um eine erste Einführung handelt und MS-DOS weitaus umfangreicher ist.
Peter Freese beginnt seine Erläuterungen mit dem Einschalten der Anlage, beschreibt die Bestandteile der Konfiguration, ihr Zusammenspiel bevor er auf wichtige Einzelheiten der Steuerung, die durch das Betriebssystem übernommen wird, eingeht. Wichtig scheint ihm auch der richtige Umgang mit den doch so anfälligen Datenträgern – den Disketten. Die Hinweise zur Datenablage sind für jeden Anwender wichtig, werden aber gern über all die vielen interessanten Details beim Arbeiten mit MS-DOS vergessen. Ein fortführender Band wäre wünschenswert.

Lore Schönrock

Einführung in MS-DOS

Von Wolfgang Schneider. Einführung in die Anwendung des Betriebssystems MS-DOS. Mit Übungsaufgaben und Lösungen, Programmieren von Mikrocomputern, Band 16, Vieweg Verlag, Braunschweig, 1985, 38 DM, 174 Seiten.
ISBN 3-528-04365-2

Wolfgang Schneider wendet sich mit seinem Buch an den Computer-Neuling. Durchstrukturiert wie eine wissenschaftliche Arbeit fällt dem Leser der Überblick nicht schwer. Nach gleichem Schema wird in jedem Kapitel vorgegangen. Zunächst erhält man einen allgemeinen Einblick in die Problemstellung, dann werden ausgewählte, wesentliche Themen detailliert behandelt. Eine Zusammenfassung mit anschließenden Übungsaufgaben beendet die Ausführungen. So können die Kapitel auch unabhängig von ihrer Reihenfolge gelesen und erarbeitet werden. Die umrahmten Befehlsfolgen sind mit einem Blick erfassbar, ebenso die eingerahmten und fettgedruckten Merksätze.
Am Ende des Buches sind die Lösungen der Übungsaufgaben zu finden sowie ein Glossary mit den wichtigsten Begriffen um den Computer, eine Beschreibung des ASCII-Codes mit Tabelle und die Umwandlung von Binär- in Dezimalzahlen. Der Autor hält den Editor EDLIN für ein so wichtiges Hilfsprogramm des Betriebssystems, daß er ihn in seinem Einführungskurs sehr ausführlich beschreibt, während er leider auf die Einführung von COPY.CON verzichtet. Da der Einsteiger aber meist nur kurze Programme schreibt, erscheint mir COPY.CON gerade in diesem Stadium verständlicher. Doch ist das Buch von Wolfgang Schneider ein empfehlenswertes Lesebuch und nützliches Nachschlagewerk für Grundbegriffe zur Handhabung des PC.

Lore Schönrock

Multiplan kompakt

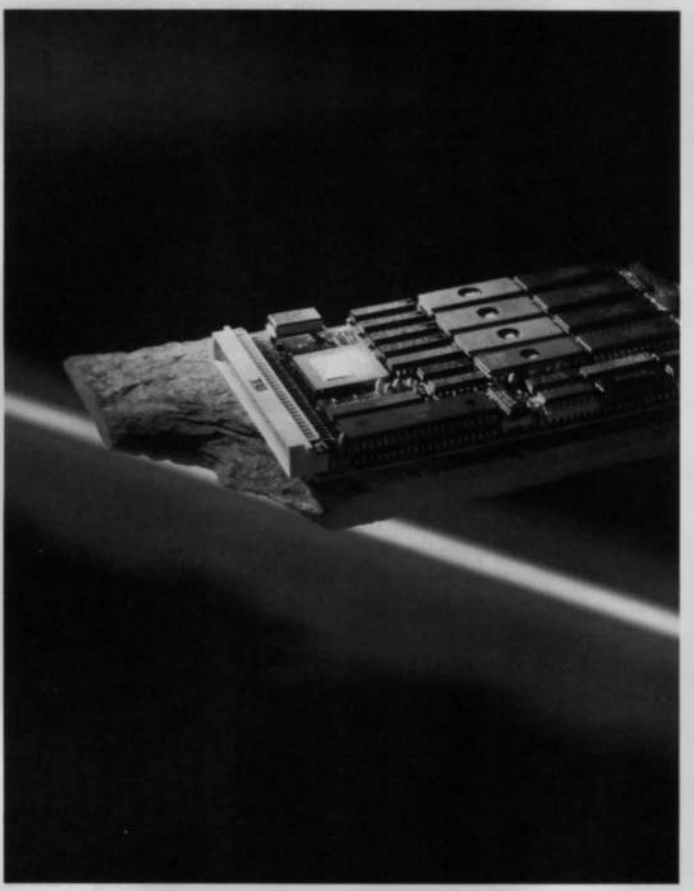
Einführung in das Tabellenkalkulations-Programm Multiplan 3.0. Von Dr. Andreas Pfeifer. 124 Seiten, 30 Abbildungen, kartoniert, 16,80 DM. Franzis-Verlag, München, 1988.
ISBN 3-7723-2201-8

An alle Anwender von Multiplan 3.0 wendet sich Dr. Andreas Pfeifer mit seinem Nachschlagewerk. Dabei zielt er besonders auf die Praktiker ab, die eine Stütze bei der Arbeit brauchen. Das handliche Format des Buches kommt der Arbeit am PC entgegen. Der Anfänger findet viele Beispiele und Hinweise, die den Einstieg in das professionelle Arbeiten mit Multiplan 3.0 erleichtern. *Rs*

Gerd Häußler

Die mc-Transputerkarte *Teil 1*

Die mc-Transputerplatine wurde entwickelt, um Ihnen einen Low-Cost-Transputersystem zur Verfügung zu stellen. Das System ist universell einsetzbar. Zum Betrieb genügt eine V.24-Verbindung. Alles andere erledigt die Software in Assembler, vom PC oder Atari aus.



Sie haben richtig gelesen. Hier können Sie den Transputer a la mc kennenlernen, Schritt für Schritt von unten herauf. Damit Sie beim Studium der Transputer nicht ins Schwimmen kommen. Seit etwa zwei Jahren geistert das Thema Transputer schon durch die Fachpresse. Da sich aber die Preise für Transputersysteme in so hohen Regionen bewegt haben, daß der Einsatz und das genaue Studium dieser neuartigen Mikroprozessoren in der Praxis oft zu teuer war, galten die Transputer zwar als intellektuelle Herausforderung aber nur in hochfliegenden Spezialaufgaben als einsetzbar. Jetzt wird das anders. Die mc-Transputerplatine zeigt, daß die Preise fallen und daß Transputer-Know-how erlernbar ist.

Die mc-Transputerplatine ist als Europakarte sowohl eine Stand-Alone-Lösung, als auch ein Modul für ein beliebig konfigurierbares Parallelprozessorsystem. Um dabei den industriellen Einsatz als High-Speed-Steuersystem zu erlauben, wurde die Platine mit statischen RAMs versehen (siehe *Bild 2*).

Der Transputer T414

Der Transputer mit der Bezeichnung T414 auf dieser Platine ist ein 32-Bit-Mikroprozessor, der eigentlich noch zur Klasse der RISC-Prozessoren gehört. Er wurde vom Design her speziell auf eine möglichst einfache Vernetzbarkeit zum Aufbau eines beliebig großen und beliebig strukturierten Parallelcomputersystems ausgelegt. Dazu gibt es auf dem Chip T414 die sogenannten Links, die bei klassischen Mikroprozessoren nicht vorkommen. Diese Links sind spezielle Hochgeschwindigkeits-Schnittstellen, die mit einer Datenrate von 5 bis 20 MBit/s arbeiten. Ein Transputer ist dabei mit vier solchen Links ausgestattet, die intern direkt über DMA-Transfer angesprochen werden (*Bild 3*).

Die zweite Neuerung, die Transputer vom sonstigen Mikroprozessormarkt abhebt, ist der interne Prozeß-Scheduler. Das ist eine Hardwareeinrichtung, die intern im Prozessor mit möglichst kleinem Zeitaufwand eine Prozeßumschaltung bewerkstelligen kann. Eine Prozeßumschaltung wird im Durchschnitt in 600 ns (!) erledigt.

Ein weiteres Merkmal modernster Mikroprozessortechnik ist das 2-KByte-RAM on the Chip. Dieser Chip-interne Speicher kann mit der sehr hohen Taktfrequenz von 20 MHz betrieben werden. Das Raffinierte daran: Die sehr hohe Taktfrequenz von 20MHz bleibt auf das Innere des Transputerchips beschränkt. Durch eine spezielle Schaltung im Transputer-Chip wird die von außen angelegte Frequenz von 5 MHz auf 20 MHz hochgeschraubt. Hochfrequenz-Probleme sind deshalb im Schaltungsaufbau nicht zu erwarten (Entwerfen Sie einmal eine Platine, auf der überall 20 MHz anliegen müssen). Bei 20 MHz chip-interner Taktfrequenz wird mit einer Zykluszeit von 50 ns auf den internen Speicher zugegriffen. Hier kann mit einer Geschwindigkeit auf einen 2 KByte großen RAM-Bereich zugegriffen werden, wie sie bei andern Prozessoren nur bei Register-zu-Register-Operationen erreicht werden kann. Bild 3 zeigt den inneren Aufbau eines Transputers im Blockschaltbild.

Die Transputer-Links

Die vier bidirektionalen, seriellen Kommunikationskanäle, die jeder Transputer besitzt, werden als INMOS-Links bezeichnet. Diese Kanäle arbeiten beim T414 mit 5, 10 oder 20 MBit/s Datenübertragungsrate, die sich am Transputer über die Hardware einstellen läßt. Die Schnittstellen sind TTL-kompatibel und lassen sich durch Verwendung von Standardtreibern (z. B. für RS 422) auch für Entfernungen im Bereich bis zu 30 Metern einsetzen. Für den Anschluß an andere Systeme wurde von INMOS ein Link-Adapter-Baustein mit der Bezeichnung C011/C012 entwickelt, der die Wandlung von parallelen Daten auf das spezielle INMOS-Linkformat (und umgekehrt) vornimmt. Die INMOS-Links werden im Transputer durch DMA-Transfer bedient. Das Handshaking mit dem Kommunikationspartner geschieht automatisch, ohne Softwarekontrolle. Der Empfänger quittiert jedes empfangene Byte durch Senden einer Bestätigung. Diese besteht aus der Folge der zwei Bit 1 und 0. Erst nach Erhalt der Empfangsbestätigung wird das nächste Byte gesendet. In der übernächsten Ausgabe von mc, nach der genauen Schilderung der Transputer-Platine, wird anhand von Programmbeispielen aufgezeigt, wie die Links von der Software angesteuert werden können. Dann wird auch näher auf die

Multiprozeß-Fähigkeit des Transputers eingegangen. Vorerst kann hier erst einmal auf den in mc, Ausgabe 8/87 erschienenen Artikel verwiesen werden. Wie die dort geschilderten Mechanismen im Detail aus der Sicht des Programmierers aussehen, wird dann an Beispiel-Programmen in Assembler gezeigt. im Zusammenhang mit den TransputerLinks ist noch ein weiteres Merkmal der Transputertechnologie zu erwähnen. Bei „normalen" Mikroprozessoren steht eine Initialisierung, bzw. ein Bootvorgang, immer mit dem Einsatz von EPROMs in Verbindung. Dies würde bei Prozessornetzwerken, bei denen viele Prozessoren zusammengekoppelt sind, Probleme bereiten. Jedem dieser Prozessoren müßte zum Beispiel ein eigenes EPROM für die Initialisierung zugeordnet sein.
Bei Transputern läßt sich die Initialisierung auch über die Links bewerkstelligen. Wenn das Signal am Pin 'BootFromROM' auf Nullpegel liegt, erwartet der Transputer nach einem Reset ein Urladerprogramm über eines seiner vier Links. Welches der Links diesen Ladervorgang übernimmt, ist egal. Es wird immer das Link als Urlader-Schnittstelle deklariert, an dem nach der Reset-Sequenz die ersten Daten ankommen. Dabei ist sichergestellt, daß sich ein beliebiges Transputernetzwerk über die verknüpften Links automatisch booten läßt (Beispiel in *Bild 4*).

Der Befehlssatz

Daß der Transputer in die Klasse der RISC-Prozessoren einzuordnen ist, wurde schon gesagt. RISC bedeutet „Reduced Instruction Set Computer". Der Befehlssatz wird bei einer solchen Maschine im Vergleich zu einer CISC-Architektur (Complex Instruction Set Computer) möglichst einfach gehalten, wobei erstens auf größtmögliche Ausführungsgeschwindigkeit und zweitens auf eine möglichst gute Anpassung an Hochsprachen Wert gelegt wird.
Bevor nun einiges Grundlegende über den Transputerbefehlssatz gesagt wird, soll zum besseren Verständnis erst einmal ein Blick auf den Registersatz eines Transputer geworfen werden (siehe *Bild 5*). Die drei Register A, B und C sind zunächst am interessantesten für den Programmierer. Über sie werden alle Berechnungen und Datenmanipulationen abgewickelt. Die drei Register sind als Stack organisiert, d.h. es kann prinzipiell immer nur auf das Register A zugegriffen werden. Der Transputer enthält also wegen seiner RISC-Architektur keine General Purpose Register, wie sie bei anderen Prozessoren üblich sind. Dies ist allerdings kein Nachteil, da ein Transputer auf

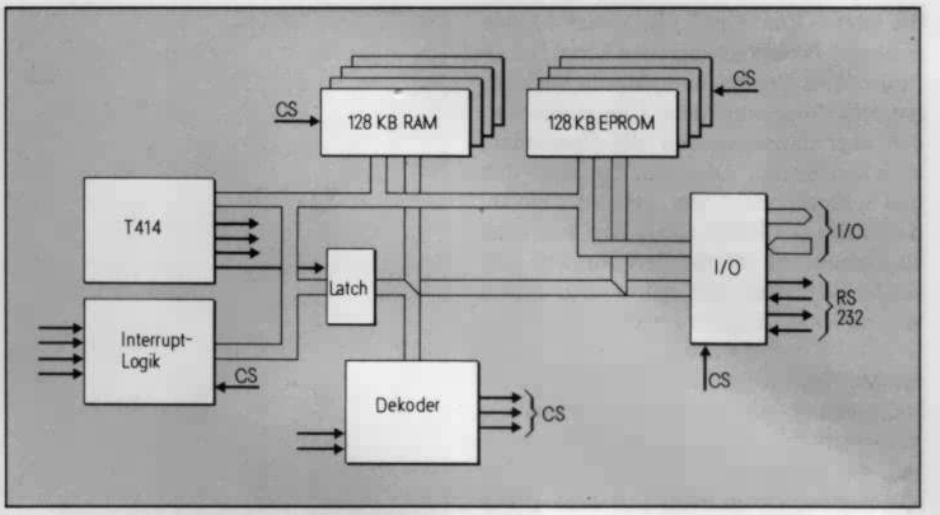


Bild 2. Das Blockschaltbild der mc-Transputerplatine TA-1

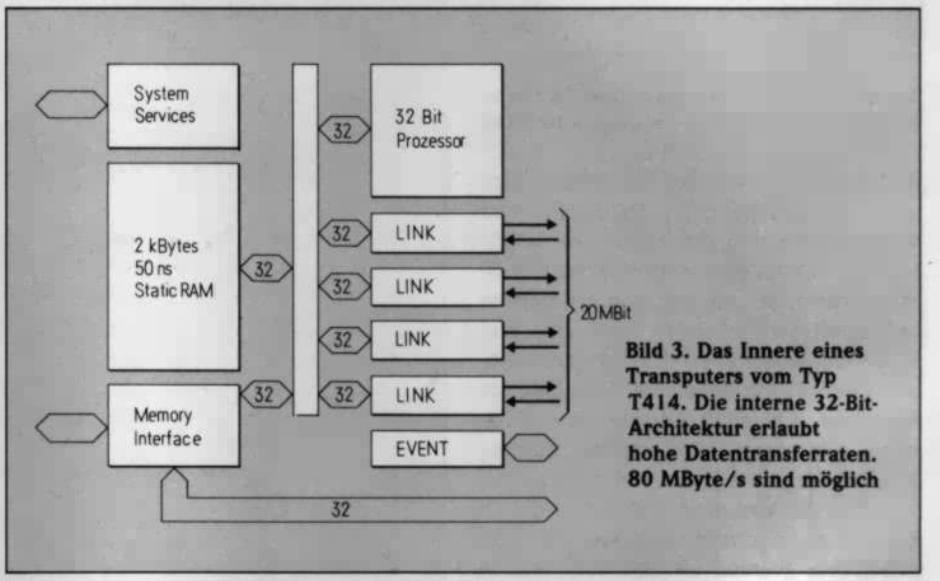


Bild 3. Das Innere eines Transputers vom Typ T414. Die interne 32-Bit-Architektur erlaubt hohe Datentransferraten. 80 MByte/s sind möglich

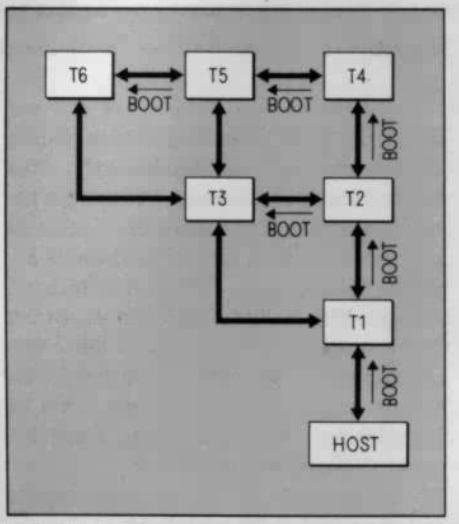


Bild 4. Ein Transputernetzwerk läßt sich über die Links vollständig booten. Man kann also auch Systeme aufbauen, die vollständig ohne EPROMs auskommen

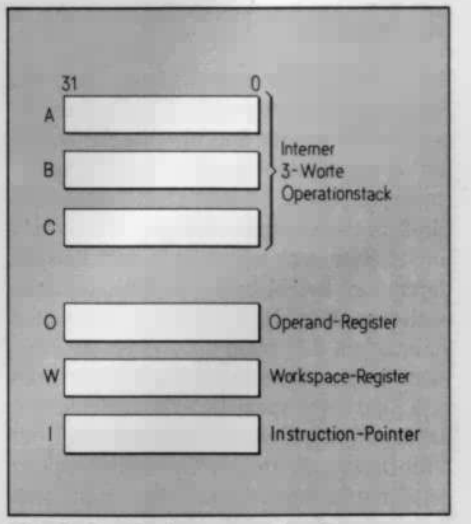


Bild 5. Der Registersatz eines Transputers. Dazu kommen noch die 2048 General-Purpose-Register auf dem Chip, die praktisch ohne Zeitverlust erreicht werden können

das interne RAM von 2 KByte oder 4 KByte in einem Taktzyklus zugreifen kann. Das O-Register, als Operandenregister bezeichnet, hat eine Doppelfunktion. Zum einen werden über dieses Register alle Operanden, also Konstanten, Adressen, Sprungweiten und Sprungadressen usw. geladen. Zum andern kann der Inhalt dieses Registers auch als Befehlscode interpretiert werden. Dieser Mechanismus wird später noch einmal anhand eines Beispiels genauer erläutert.

Ein Maschinenbefehl eines Transputers besteht immer aus einem Byte, wobei die höherwertigen 4 Bit die Operation codieren, die niederwertigen 4 Bit für die Ansprache der Daten reserviert sind (siehe

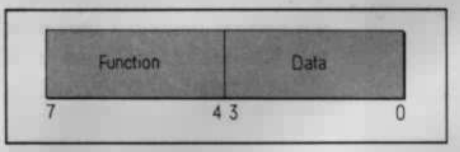


Bild 6. So ist ein Code-Byte aufgebaut. Je vier Bit bilden den Funktions- und den Adreß-Teil

Bild 6). Dies scheint auf den ersten Blick sehr eng zu sein, denn 16 Befehle und 4-Bit-Datenworte sind wohl für eine 32-Bit-Maschine etwas dürftig. In Wirklichkeit ist es jedoch so, daß die dreizehn am meisten benötigten Befehle direkt durch eine Ein-Byte Codesequenz erreicht werden können. Diese 13 Grundbefehle, die mit einem Argument, d.h. einer Adresse, einer Konstante oder einem Offset versorgt werden, erhalten durch die unteren vier Bit im Befehlsbyte ihr Argument, welches in das O-Register geladen wird. Falls vier Bit nicht ausreichen, werden sie mit dem letzten 4-Bit-Teil ihres Argumentes gefüllt.

Ein Jump

Der Befehl 'jump' zum Beispiel, dem der Funktioncode 0 zugeordnet ist, hat als Argument die relative Sprungweite zur aktuellen Programmzählerstelle. Ein Vorwärtssprung um bis zu 15 Byte kann durch einen Ein-Byte-Code erreicht werden. Ein Sprung um 10 Byte nach vorne kann zum Beispiel durch das Befehlsbyte 0A (Hexadezimal) codiert werden. Um größere Sprungweiten zu erzielen, d.h. mehr als vier Bit als Argument in das Operanden-Register zu bringen, sind zwei spezielle Präfixbefehle vorhanden. Diese Befehle bewirken, daß der 4-Bit-Datencode in das Operandenregister geladen und dieses danach um vier Bit nach links geschoben wird. Um auch negative Zahlen einfach zu generieren, existiert der 'Negativ-Präfix'-Befehl, der alle Bits im Operandenregister komplementiert, bevor sie um die 4 Bit geschoben werden. Durch mehrere solcher Präfixbefehle kann man den gewünschten Operanden sozusagen zusammenbasteln. Sind dabei große Zahlen zu generieren, werden mehr Präfixbefehle benötigt, bei kleineren Zahlen entsprechend weniger. Wie in *Bild 7* zu sehen ist, ist der Operationsteil des Präfix-Befehls mit 2 verschlüsselt. Um bei dem Sprungbefehl als Beispiel zu bleiben, wird ein Sprung um 24 (hexadezimal) Byte nach vorne durch folgende Codesequenz aufgebaut:
22 – Präfixbefehl, 2 ins Operandenregister und dann Shift 4 Bit nach links.
04 – Code für jump, 4 ins Operandenregister, dieses enthält jetzt 24 (Hex), danach Ausführung des Sprunges nach vorne.

Direkte Funtionen

Funktionscode	Codesequenz	Mnemonischer Name	Prozessorzyklen	Bedeutung
0	0X	j	3	jump [D]
1	1X	ldlp	1	load local pointer
2	2X	pfix	1	prefix
3	3X	ldnl	2	load non-local
4	4X	ldc	1	load constant
5	5X	ldnlp	1	load non-local pointer
6	6X	nfix	1	negative prefix
7	7X	ldl	2	load local
8	8X	adc	1	add constant [E]
9	9X	call	7	call
A	AX	cj	2	conditional jump (not taken) contitional jump (taken)
B	BX	ajw	1	ajust workspace
C	CX	eqc	2	equals constant
D	DX	stl	1	store local
E	EX	stnl	2	store non-local
F	FX	opr	-	operate

Bild 7. Die 16. Grundbefehle des Transputers, die mit 4 Bit „Datenteil" versorgt werden können

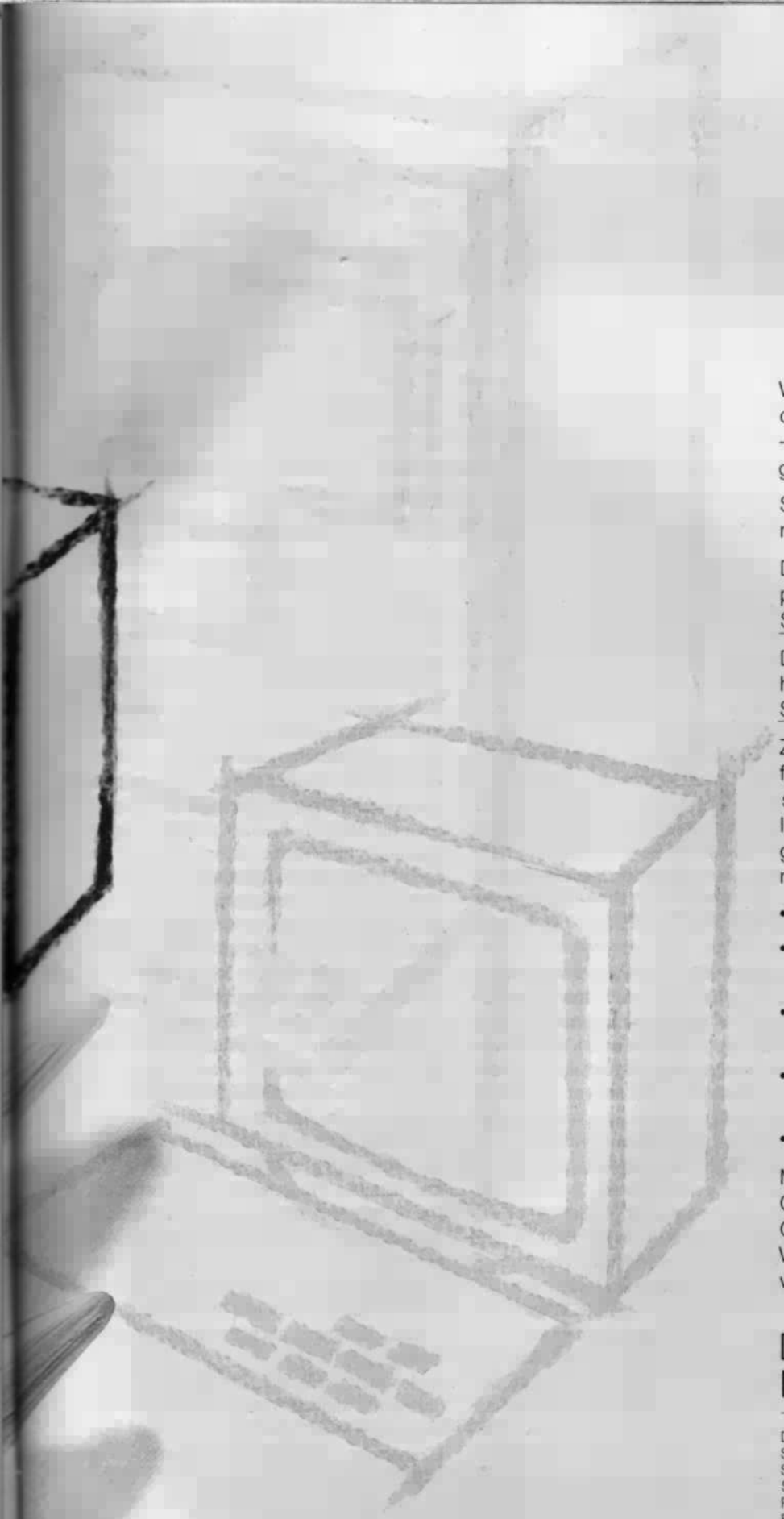
Wie der Aufbau eines komplexeren Befehls abläuft, soll kurz an einem zweiten Beispiel erläutert werden. Angenommen, der Befehl 'and', dem der Hexcode 46 zugeordnet ist, soll ausgeführt werden. In diesem Fall wird durch den Präfixbefehl 24, das High Nibble des Befehlswortes in das Operandenregister geladen. Bis hierher unterscheidet sich dies in keinster Weise vom Laden eines Operanden. Als nächstes wird mit der Opr-Instruktion F6 das Low Nibble des Befehlscodes der and-Instruktion in das Operandenregister geladen und gleichzeitig der Befehl ausgeführt. Die Opr-Instruktion bewirkt nämlich, daß der Inhalt des Operanden-Registers als Befehl und nicht als Datenwort interpretiert wird. So gesehen wird

Code	Size	Abbreviation	Cycles	
#00	short	rev	1	reverse
#05	short	add	1	add
#0C	short	sub	1	subtract
#53	long	mul	bpw+6	multiply
#2C	long	div	bpw+7	divide
#1F	long	rem	bpw+5	remainder
#52	long	sum	1	sum
#04	short	diff	1	difference
#08	short	prod	b+4	product
#46	long	and	1	and
#4B	long	or	1	or
#33	long	xor	1	exclusive or
#32	long	not	1	bitwise not
#41	long	shl	n+2	shift left
#40	long	shr	n+2	shift right
#09	short	gt	1	greater than
#21	long	lend	10/5	loop end - loop back/exit loop
#34	long	bcnt	2	byte count
#3F	long	wcnt	5	word count
#1B	long	ldpi	2	load pointer to instruction
#42	long	mint	1	minimum integer
#02	short	bsub	1	byte subscript
#0A	short	wsub	2	word subscript
#4A	long	move	2w+8	move message
#07	short	in	2w+19	input message
#0B	short	out	2w+19	output message
#01	short	lb	5	load byte
#3B	long	sb	5	store byte
#0E	short	outbyte	23	output byte
#0F	short	outword	23	output word
#06	short	gcall	3	general call
#3C	long	gajw	2	general adjust workspace
#20	long	ret	5	return
#0D	short	startp	12	start process
#03	short	endp	13	end process
#39	long	runp	10	run process
#15	long	stopp	11	stop process
#1E	long	ldpri	1	load current priority
#22	long	ldtimer	2	load timer
#2B	long	tin	?	timer input
#43	long	alt	2	alt start
#44	long	altwt	?	alt wait
#45	long	altend	4	alt end
#4E	long	talt	4	timer alt start
#51	long	taltwt	?	timer alt wait
#49	long	enbs	3	enable skip
#30	long	diss	4	disable skip
#48	long	enbc	7*	enable channel
#2F	long	disc	8	disable channel
#47	long	enbt	8	enable timer
#2F	long	dist	?	disable timer
#13	long	csub0	2	check subscript from 0
#4D	long	ccnt1	3	check count from 1
#29	long	testerr	3*	test error false and clear
#55	long	stoperr	2	stop on error
#10	long	seterr	1	set error
#3A	long	xword	4	extend to word
#56	long	cword	5	check word
#1D	long	xdble	2	extend to double
#4C	long	csngl	3	check single
#16	long	ladd	2	long add
#38	long	lsub	2	long subtract
#37	long	lsum	2	long sum
#4F	long	ldiff	2	long diff
#31	long	lmul	bpw+1	long multiply
#1A	long	ldiv	bpw+3	long divide
#36	long	lshl	n*	long shift left
#35	long	lshr	n*	long shift right
#19	long	norm	bpw+5*	normalise
#12	long	resetch	3	reset channel
#2A	long	testpranal	2	test processor analysing
#18	long	sthf	1	store high priority front pointer
#1C	long	stlf	1	store low priority front pointer
#54	long	sttimer	1	store timer
#50	long	sthb	1	store high priority back pointer
#17	long	stlb	1	store low priority back pointer
#3E	long	saveh	4	save high priority queue registers
#3D	long	savel	4	save low priority queue registers
#57	long	clrhalterr	1	clear halt-on-error
#58	long	sethalterr	1	set halt-on-error
#59	long	testhalterr	2	test halt-on-error
#72	long	fmul	40	fractional multiply
#63	long	unpacksn	16	unpack single length fp number
#6D	long	roundsn	15	round single length fp number
#6C	long	postnormsn	30*	post-normalise correction of single length fp
#71	long	ldinf	1	load single length infinity
#73	long	cflerr	3	check single length fp infinity or NaN

Bild 8. Die Maschinenbefehle des T414 im Überblick. Der unter 'Code' angegebene Hexwert ist der Befehlscode, der im Operanden-Register bei der Ausführung des Opr-Befehls stehen muß. Die Timing-Angaben beziehen sich auf den Speicherzugriff im internen RAM. Die mit '*' markierten Timing-Angaben sind als worst-case zu betrachten. Die mit '?' markierten Befehle, wie z. B. Befehle, die die Timer-Warteschlange betreffen, sind von äußeren Umständen abhängig. Deshalb kann natürlich keine Angabe über die Ausführungszeit dieser Befehle gemacht werden

SIEMENS

Die Welt der IndustriePC ist farbiger geworden



Wenn Ihre Welt die Fertigung ist, oder das Technische Büro, oder das Labor – dann haben wir heute gute Nachrichten für Sie:

Speziell für Ihre „Welt" gibt es nun zwei neue IndustriePC!

Der eine ist besonders kompakt und preisgünstig und heißt SICOMP® PC 16-15.

Der andere ist superschnell, hat 32-Bit-Architektur und heißt SICOMP PC 32-05.

Zusammen mit dem bewährten, vielfältig ausbaufähigen SICOMP PC 16-20 sind sie eine voll AT-kompatible IndustriePC-Familie. Mit außergewöhnlichen gemeinsamen Merkmalen:

- Schutzart nach Maß
- mehrere Realtime-Betriebssysteme zur Wahl
- Kommunikationsfähigkeit im Automatisierungsverbund
- extrem hohe Funktionssicherheit in Ihrer „Welt" und
- aufgabengenaue Projektierbarkeit.

Mehr Informationen?
Gern!
Greifen Sie zum Stift.
Wir senden Ihnen postwendend das Info-Paket.

„progress in automation: Siemens"

Die IndustriePC Natürlich von Siemens

Das kostenlose Info-Paket über die IndustriePC liegt für Sie bereit.
Schreiben Sie an
Siemens AG, Infoservice 215/Z 314 k
Postfach 2348, 8510 Fürth.
Btx: ✻ 32000212614 #

Name

Firma/Abteilung

Straße/Tel.

PLZ/Ort

A 19100 - E 851 - Z 833

Vergleich der Whetstone-Resultate

System	Leistung bei einfacher Genauigkeit (in 1000 Whetstones/s)
IMS T800	4000
WE 32200/32206-24	2800
VAX 11/780	1083
MVII	925
SUN-3	860
NS 32332/32081	728
IMS T414-20	667
NS 32032/32081	390
i 80286/80287	300
IBM RT-PC/FPA	200
i 8086/8087	178
MC 68000	13
IBM RT-PC	12

Bild 9. Die Rechenleistung von Einzel-Transputern in Whetstones, verglichen mit verschiedenen anderen Prozessoren (teilweise mit Coprozessor). Transputernetzwerke können mit der Rechenleistung in bestimmten Fällen an die Supercomputer herankommen

der and-Befehl durch die Codefolge 24F6 generiert (*Bild 8* zeigt alle T414-Befehlscodes im Überblick). Da der Transputer, wie vorhin schon erwähnt, intern auf einer dreielementigen Stackarchitektur aufbaut, wirkt dieser and-Befehl auf die Register A und B. Das Ergebnis der Operation wird wiederum in A abgelegt, das Datum in C rutscht hoch in Register B und das Register C ist undefiniert.

Dieser Überblick muß für einen ersten Eindruck der Transputer-Assemblersprache genügen. In der übernächsten Ausgabe werden wir speziell auf das Thema Software eingehen und auch Programmbeispiele für den im Lieferumfang der TA-1-Karte enthaltenen Assembler aufzeigen. Dann kann sich jeder selbst davon überzeugen, daß die eingangs erwähnten Benchmarks der Realität entsprechen. Für alle, die diese Zahlen noch nicht kennen, sind sie in *Bild 9* noch einmal zusammengestellt.

Das mc-Einplatinen-Transputersystem

Das mc-Einplatinen-Transputersystem ist speziell für den Einsatz in rechenaufwendigen Steuersystemen konzipiert, wobei besonderer Wert auf eine preisgünstige Lösung gelegt wurde. Wegen des günstigen 'Preis/Mips-Verhältnisses' kann diese Karte sowohl im praktischen Einsatz als auch als Trainings- oder Evaluation-System verwendet werden. Der Einstieg in die Transputertechnologie kann so von all denen gewagt werden, die bis jetzt von den recht hohen Preisen abgeschreckt wurden.

Um für den Industrieeinsatz genügend Sicherheit zu gewährleisten, werden statischen RAMs eingesetzt. Dabei sind Speichergrößen von 128 KByte EPROM und 128 KByte RAM sowie (bei Verwendung des Systems als Subsystem) auch von 256 KByte RAM auf einer Platine vorgesehen. Es werden Standardspeicherbausteine mit 32 KBit × 8 -Organisation verwendet. Der I/O-Chip auf der Platine ist der Typ MC 2681 von Motorola. Er besitzt zwei serielle Schnittstellen, die über einen MAX-238-Baustein auf die VG-Leiste herausgeführt sind. Es stehen sieben parallele Mehrzweckeingänge und acht parallele Mehrzweckausgänge zur Verfügung. Beim Einsatz der Karte als Steuerrechner werden die auf der Karte angebotenen Schnittstellen sicherlich in vielen Fällen nicht ausreichen. Deshalb sind über die 96polige VG-Leiste neben den Ein- und Ausgängen des I/O-Bausteins MC-2681 alle T414-Bussignale herausgeführt. Damit ist gewährleistet, daß vom Anwender beliebige Zusatzschaltungen, respektive I/O-Bausteine oder zusätzliches RAM, angeschlossen werden können. Da die Karte auch ohne PALs auskommt und neben dem Transputerchip keine außergewöhnlichen Bauelemente enthält, ist die Karte auch beherrschbar, wenn das Entwicklungslabor nur eine bescheidene Ausstattung besitzt. Ehe es in der nächsten mc-Ausgabe in Bezug auf die Hardware ans Eingemachte geht – dort wird der TA-1 in allen Einzelheiten beschrieben – noch ein paar Worte zur ...

... Softwareentwicklung

Für die Softwareentwicklung auf dem Transputerboard TA-1 sind verschiedene Varianten vorgesehen. Zum einem wird zur Karte ein Crossassembler und Linker, der auf einem IBM-PC/XT/AT ablauffähig ist, mitgeliefert. Dabei ist im EPROM ein einfaches Monitorprogramm mit Download-Möglichkeiten vorhanden. Software kann dann auf dem Host entwickelt und über eine serielle Schnittstelle in die Karte geladen werden. Diese Möglichkeit ist übrigens auch für Atari-ST- Besitzer gegeben: Crossassembler und Terminalemulation werden auch für den Atari angeboten werden. Ein weiterer Weg, die so erzeugte Software in die Karte zu laden, ist der Einsatz einer Linkadapterkarte, die als PC-Einsteckkarte verfügbar ist. In diesem Fall kann die mc-Einplatinen-Transputerkarte ohne EPROMs mit vollem RAM-Ausbau betrieben werden.

Für Hochsprachenentwicklungen sind zwei Möglichkeiten gegeben: Zum einen wird ein C-Crosscompiler für den IBM-XT/AT erhältlich sein, der mit dem mitgelieferten Assembler-System in Verbindung steht. Dabei wird durch spezielle Library-Funktionen parallele Programmierung unterstützt. Damit werden dann auch echte Parallel-Anwendungen ermöglicht. Durch die Vernetzung von mehreren mc-Transputer-Boards kann eine Rechenleistung erreicht werden, die im Moment nur durch genau solche Rechner hochgerechnet werden kann ...

Andererseits läßt sich die Karte auch als Subsystem anderer Transputersysteme verwenden, wobei hier auch auf das TDS (Transputer Development System) von INMOS, welches einen OCCAM-Compiler enthält, verwiesen werden kann. Mit diesem System läßt sich die Karte als Komponente eines beliebigen Transputernetzwerkes einsetzen. Anwendern die im Moment schon Transputer einsetzen, wird hiermit die Möglichkeit einer günstigen Erweiterung ihres Systems geboten.

Wird fortgesetzt

Erich Esders

Arbeiten mit dem 80287

Teil 1

Hier soll mit konkreten Programmen in Assembler und C vorgeführt werden, wie man den 80287 in neuen oder bestehenden Programmen einsetzt und was das für Nutzen und Vorteile besitzt. Der 80287 kostet fast so viel, wie eine ganze AT-Hauptplatine. Aber er lohnt sich für alle, die viel rechnen müssen.

Der 80287 ist ein sogenannter Coprozessor zum 80286. Er unterstützt den 80286-Prozessor, indem er Rechenarbeiten übernimmt, für die der Hauptprozessor umständlich und lang programmiert werden müßte. Diese Rechenarbeiten werden im 80287 direkt von der Hardware superschnell erledigt. Die einzige „Rechenaufgabe" für den 80286 besteht dann noch in der Verwaltung der Operanden, der Operatoren sowie der Resultate der Floating Point Unit. Weil beide Prozessoren so aufeinander abgestimmt sind, daß auch diese Verwaltung flottest geschieht, bilden beide Prozessoren ein kräftiges Gespann.

Der IEEE 754 Floating Point Standard

Betrachten Sie folgendes Problem: In einem Programm trete während der Berechnung eines Bruches ein Ausdruck „X-Y" auf, der die „Reals" X und Y enthält und der später als Divisor genutzt werden soll. Um sicher zu gehen, daß im Nenner keine 0 auftreten wird, vergleicht das Programm vor der Division die beiden Variablen. Nur bei Ungleichheit darf das Programm die Division durchführen. Trotzdem tritt „manchmal" ein „Division by zero"-Fehler auf, und zu allem Übel ist dieses „manchmal" auf unterschiedlichen Rechnern auch noch unterschiedlich häufig! Der Grund ist in der rechnerinternen Darstellung der reellen Zahlen zu suchen.

In der Mathematik liegen die reellen Zahlen dicht auf der Zahlengeraden, es gibt keine Lücke zwischen ihnen. Zwischen zwei Nachbarn auf der Zahlengraden läßt sich immer noch ein dazwischen liegender Wert ermitteln. Für die genaue Darstellung der meisten reellen Zahlen im Rechner wären deshalb unendlich viele Bit pro Zahl notwendig. Wegen der Begrenzung der Anzahl der verwendeten Bit zur Darstellung (normalerweise Vielfache eines Bytes) von Zahlen kann in einer Maschine der größte Teil der reellen Zahlen nicht dargestellt werden. Es war und ist ein großes Problem für die Systemprogrammierer, zu einer vorgegebenen reellen Zahl immer die nächstgelegene Maschinenzahl zu finden.

In den USA hatte in den 70er Jahren das „Institute of Electrical and Electronics Engineers, Inc." oder kurz IEEE (auch leger „Ei-Tripl-I" ausgesprochen!) eine Arbeitsgruppe gebildet, mit der Aufgabe, eine Standardisierung für die rechnerinterne Darstellung und die Implementation zur Arithmetik von Zahlen in Mikrocomputern zu entwickeln. Das Resultat dieser Bemühungen ist der sogenannte IEEE 745 Floating Point Standard, der mittlerweile weltweit akzeptiert wird. Die folgenden Abschnitte enthalten eine grobe Übersicht der einzelnen Punkte dieses Standards; tiefergehende Informationen geben die daran anschließenden Erläuterungen zum Numerik-Chip selbst.

Formate

Wie *Bild 1* zeigt, unterscheidet IEEE 745 zwei Grundformate zur Darstellung reeller Zahlen im Computer: „single precision" mit einer Länge von 32 Bit und „double precision" mit einer Länge von 64 Bit. Das 32-Bit-Format teilt sich danach in ein 1-Bit-Vorzeichen s, einen 8-Bit offset-behafteten Exponenten e und eine 23-Bit lange Mantisse f ein. Exponenten werden grundsätzlich mit einem Offset versehen: Die Addition einer geeignet gewählten Konstanten (bias, zum Beispiel 127) sorgt dafür, daß der Wert des Exponenten bei der Darstellung in der Maschine (biased exponent) innerhalb seines Zahlenbereichs niemals negativ wird! Im 64-Bit-Format stehen das 1-Bit-Vorzeichen, ein 11-Bit offset-behafteter Exponent und eine 52-Bit lange Mantisse zur Verfügung. Die *Bilder 2 und 3* enthalten eine Aufstellung über die vereinbarte Bedeutung bestimmter Werte in beiden Formaten. Die Tabellen zeigen deutlich,

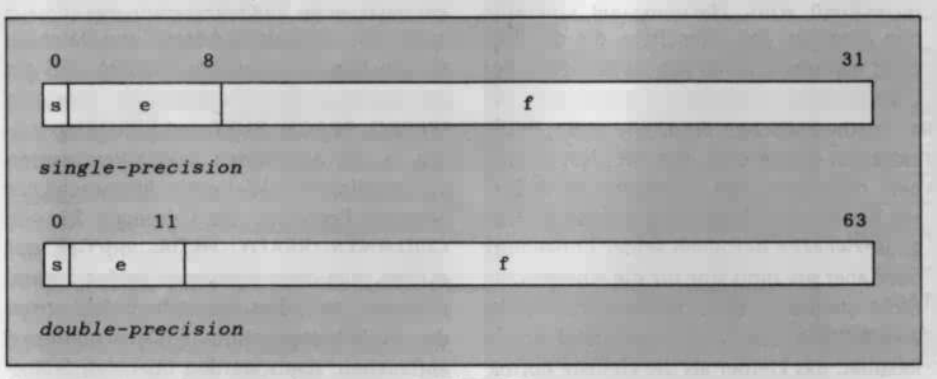


Bild 1. Die Datenformate im Standard nach IEEE 745

e	f	dargestellter Wert
255	!0	Not a Number, NaN
255	0	$(-1)^s * \infty$
1...254		$(-1)^s * 2^{e-127} * (1.f)$
0	!0	$(-1)^s * 2^{-126} * (0.f)$
0	0	$(-1)^s * 0$, Null

Bild 2. Besondere Werte in der Maschine haben verschiedene Bedeutungen. Hier für single precision

e	f	dargestellter Wert
2047	!0	Not a Number, NaN
2047	0	$(-1)^s * \infty$
1...2046		$(-1)^s * 2^{e-1023} * (1.f)$
0	!0	$(-1)^s * 2^{-1022} * (0.f)$
0	0	$(-1)^s * 0$, Null

Bild 3. Die besonderen Werte bei double precision. Die Werte signalisieren Ausnahmesituationen

daß nicht jede mögliche Bit-Kombination zur Darstellung einer reellen Zahl genutzt wird, sondern spezielle Konstellationen des Exponenten und der Mantisse der Anzeige besonderer Werte dienen! Die dritte Zeile einer jeden Tabelle zeigt eine „normalisierte" Zahl, bei der das signifikante, am weitesten links stehende Bit der Mantisse eine „1" ist und als direkt hinter dem Dezimalpunkt stehend zu interpretieren ist. Diese Normalisierung der darzustellenden reellen Zahl erreicht man durch einen Rechts- bzw. Links-Shift des Dezimalpunktes bei gleichzeitiger Inkrementierung bzw. Dekrementierung des Exponenten; die durch den Shift vollzogene Division oder Multiplikation mit 2 wird solange durchgeführt, bis die Mantisse gerade kleiner 1 geworden ist. Die 4-Byte-Struktur (32 Bit) wurde gewählt, weil sich die Genauigkeit der damit darstellbaren Zahlen für viele Fälle als ausreichend herausgestellt hatte. Wie noch weiter unten gezeigt, gestattet diese kompakte Form neben der Mantisse die Mitführung eines offset-behafteten 8-Bit-Exponenten, der die Berechnung des Reziprokwertes aller normalisierten Zahlen ohne Überlauf gestattet. Die 8-Byte-Struktur ermöglicht das Arbeiten in einem recht großen Zahlenbereich. Zusätzlich wurde hier ein Exponentenformat gewählt, das es gestattet, acht 32-Bit-Zahlen miteinander zu multiplizieren, ohne daß dabei ein Überlauf entsteht. Weiterhin legt der Standard Anforderungen an die temporäre Zahlendarstellung fest, die mit ihrer über die 64 Bit hinaus erweiterten Genauigkeit für die korrekte Weiterverarbeitung von Zwischenergebnissen sorgen soll.

Operation	s	e[1]	f
Wahres Resultat	0	-129	1.01100110000...00
Denormalisieren	0	-128	0.10110011000...00
Denormalisieren	0	-127	0.01011001100...00
Denormalisieren	0	-126	0.00101100110...00
Denormal. Resultat[2]	0	-126	0.00101100110...00

Bild 4. Die Denormalisation

1 Der Exponent e ist ohne Offset als Dezimalzahl angegeben!
2 Vor dem Speichern wird f auf 24 Bit gerundet und zu e 126 addiert!

Arithmetik

IEEE 745 fordert für alle Berechnungen eine Genauigkeit von +/- einem halbem Bit in der letzten Stelle des gewählten Formats. Zu diesem Zweck sind in den „Zwischenrechnungen" besondere Bits zu führen, die beispielsweise Auskunft über die gewünschte Rundung geben. Eine Rundung soll ein temporäres Resultat in eines der beiden Floating-Point-Formate oder gar in das INTEGER-Format überführen. Hier ist für Arithmetik-Programme mindestens gefordert, daß die Rundung zum nächstliegenden, im Zielformat darstellbaren Wert korrekt durchgeführt wird. Liegt dabei der zu rundende Wert genau zwischen zwei darstellbaren, so ist der geradzahlige (LS-Bit = 0!) zu wählen! Der Standard schreibt weiterhin die Implementierung der vier Grundrechenarten sowie die Berechnung der Modulo-Funktion, der Quadratwurzel sowie die Konvertierungen zwischen den REAL- und INTEGER-Formaten vor. Die Modulo- oder Rest-Funktion wurde aufgrund ihres hilfreichen Einsatzes bei der Argumentenreduzierung transzendenter Funktionen übernommen. Sie transformiert beliebig große Operandenwerte einfach in das zugelassene Intervall. Die Quadratwurzel wird sehr häufig in trigonometrischen Ausdrücken, irrationalen Funktionen und Lösungsformeln von Gleichungssystemen benutzt. Ihre direkte Verwendbarkeit trägt damit wesentlich zur Vereinfachung der Programmierung bei.

Fehlersituationen und Formatausnahmen

Der Standard behandelt selbstverständlich auch Resultate, deren Wert außerhalb des Bereichs normalisierter Floating-Point-Zahlen liegt (s. a. Bilder 2 und 3). Das Eintreten eines solchen Falles wird auch mit „Exception" bezeichnet. Ein Überlauf (overflow) oder eine Division durch 0 (divide by zero) können im Computer unter bestimmten Umständen sogenannte Unendliche erzeugen (wenn man zum Beispiel qualitativ weiterrechnen will). Nachfolgende Operationen erzeugen dann Resultate, die die üblichen mathematischen Regeln bezüglich der Unendlichkeit einhalten. Operationen ohne mathematischen Sinn, wie z. B. „0/0", erzeugen ein Resultat, das mit „Not a Number" oder kurz NaN bezeichnet wird. Dieses Resultat ist eindeutig und kann zu Diagnosezwecken bezüglich seiner Entstehung oder aber als Indikator für die Anwendung nicht standardisierter Formate verwendet werden. Ein Unterlauf (Underflow) ist ein Resultat, das kleiner als die kleinste normalisiert darstellbare Zahl ist und wird nach IEEE 745 durch die Definition der „denormalisierten" Zahlen charakterisiert. Eine denormalisierte Zahl ist eine von 0 verschiedene Zahl, die zwischen der größten negativen und der kleinsten positiven, normalisierten Zahl liegt. Auch im 80287 voll implementiert, ergibt sich durch die Lösung von der Normalisierung ein weiterer Bereich von Zahlen, deren höchstwertiges Mantissen-Bit eine „0" und deren Exponent den kleinstmöglichen Wert führt. Wie in *Bild 4* gezeigt, tritt die erste „1" nach entsprechender Justage des Exponenten an einer weiter rechts gelegenen Position auf. Im Bereich der denormalisierten Zahlen kann man wieder vernünftig arbeiten. Der Unterlauf tritt nach IEEE 745 also nicht abrupt sondern graduell auf! Eine Zahl mit gültigem Exponenten, deren höchstwertiges Mantissen-Bit und eventuell noch folgende Bits gewollt „0" sind, nennt man unnormalisiert. Speziell bei der Berechnung extrem kleiner Werte kann dieses Format auf Kosten der Genauigkeit (die Mantisse wird ja nur teilweise ausgenutzt!) Anwendung finden. Eine tiefergehende Erörterung dieser Thematik würde den Rahmen der vorliegenden Arbeit sprengen; grundsätzlich ist sie ohnehin nur für denjenigen relevant, der spezielle Exception- bzw. Fehler-Handler schreiben muß (weitere Informationen s. [1]).

Das Chip-Interface

Intel bezeichnet den 80287 als Numeric Processor Extension, kurz NPX genannt. Er ist nach dem Blockdiagramm aus *Bild 5* aufgebaut; mit seinen Adress- und Datenleitungen liegt er parallel zur 80286-CPU am gemeinsamen Bus. Vier spezielle Leitungen (PEREQ, PEACK, BUSY und ERROR) gestatten es, die Aktivitäten beider Prozessoren zu koordinieren. Weiterhin überwacht der Numerik-Prozessor die Leitungen $\overline{S1}$, $\overline{S0}$, COD/$\overline{INTA}$, $\overline{READY}$, HLDA und CLK des 80286. Alle diese Leitungen bilden zusammen den Status-Bus. Wenn im Befehlsstrom des 80286 sogenannte ESC-Instruktionen auftauchen, dann werden über den Status-Bus Signale für den 80287 gesetzt, die signalisieren, daß er tätig werden soll. Das Blockdiagramm zeigt die Teilung des NPX in zwei Einheiten. Ein Teil ist die Bus Interface Unit „BIU" , der andere die Numeric Execution Unit „NEU". Voneinander unabhängig arbeitend, ist letztere für die gesamten arithmetischen Aufgaben des Chips zuständig, während die BIU die Kopplung der NEU mit der Außenwelt darstellt. Sie empfängt und dekodiert Instruktionen, organi-

siert den Operanden-Transfer mit dem externen Speicher und führt auch die Prozessor-Steuerinstruktionen aus. Die NEU enthält einen Stack aus 8 Registern mit einer Breite von 80 Bit. An diesen Stack sind über einen Exponenten- und einen Mantissen-Bus die gleichnamigen Module sowie ein Satz temporärer Register angeschlossen. Alles in allem ist damit der interne Datenbus der NEU 84 Bit breit (68 Mantisse, 15 Exponent, 1 Sign), woraus sowohl hohe Genauigkeit als auch hohe Geschwindigkeit resultieren.

Eine ESC-Instruktion ist wie in Bild 6 aufgebaut und läßt durch die fünf führenden Bits erkennen, daß es sich um eine für den Numerik-Prozessor bestimmte Anweisung handelt. Die Instruktion selbst ist in den restlichen 11 Bit dekodiert, dem kompletten Wort folgen eventuell benötigte Parameter in weiteren Byte. Der 80287 führt die einzelnen Numerik- bzw. ESC-Instruktionen nacheinander unabhängig von der CPU aus. Trifft der Prozessor 80286 in seinem Code auf eine ESC-Instruktion, so testet er in den meisten Fällen zuerst das BUSY-Signal und verzögert solange, bis der NPX frei ist. Dieser übernimmt nun seinerseits die zugehörigen Programm-Byte zur weiteren Bearbeitung während der 80286 schon im Programm fortfährt. 8086-Systeme benötigen grundsätzlich eine explizite WAIT-Instruktion zur Ausführung des eben beschriebenen Tests, in 80286-Systemen sind sie erlaubt aber nicht unbedingt notwendig. Allerdings muß bei Speicherzugriffen des NPX immer ein WAIT eingebaut werden. Wie noch weiter unten in den Beispiel-Programmen beschrieben, können beim Fehlen eines WAITs an der „richtigen" Stelle die berühmten „Murphy-Manchmals" auftreten! Aufgrund seiner Architektur kann der Programmierer den 80287 als eine einfache Erweiterung des 80286 betrachten; alle Aktivitäten zwischen beiden Prozessoren werden auf Hardware-Ebene automatisch abgehandelt. Für die Kommunikation zwischen beiden Chips sind die I/O-Ports mit den Adressen 0F8H bis 0FFH reserviert. Operationen über diese Adressen wickeln die Bausteine automatisch ab und sind von denen, die programmgesteuert erfolgen, zu unterscheiden. Um jedoch eine korrekte Funktion des NPX zu gewährleisten, darf die Software keine zusätzlichen I/O-Operationen über diese Adressen abwickeln!

Die Chip-Architektur

Bild 7 zeigt den Registersatz, so wie er sich dem Anwender darstellt. Die 80 Bit eines jeden der 8 Stackregister R1 bis R8 teilen

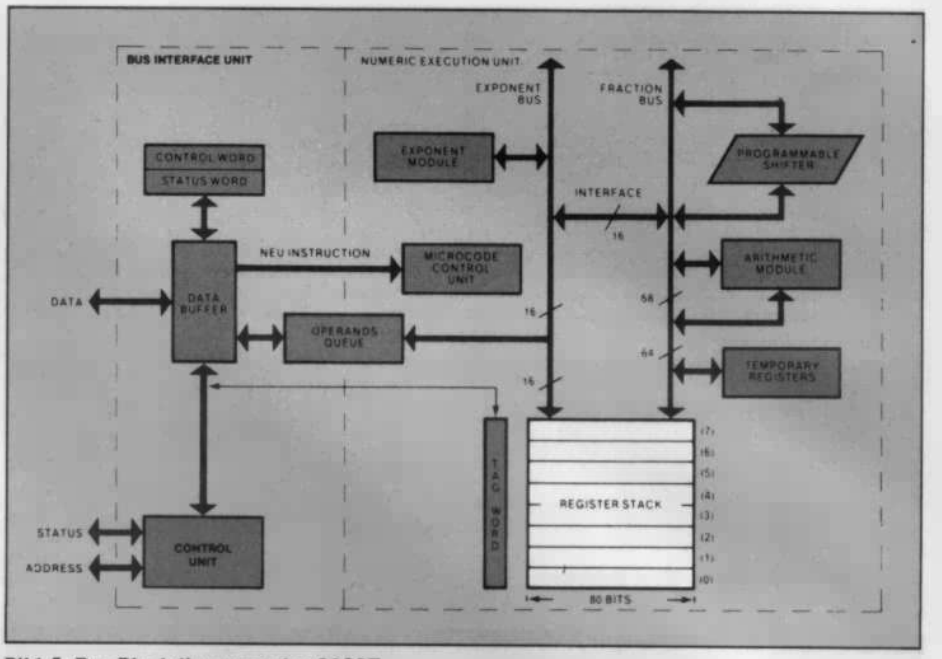


Bild 5. Das Blockdiagramm des 80287

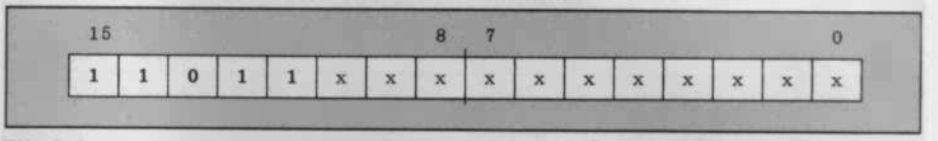


Bild 6. So ist eine NPX-Instruktion aufgebaut

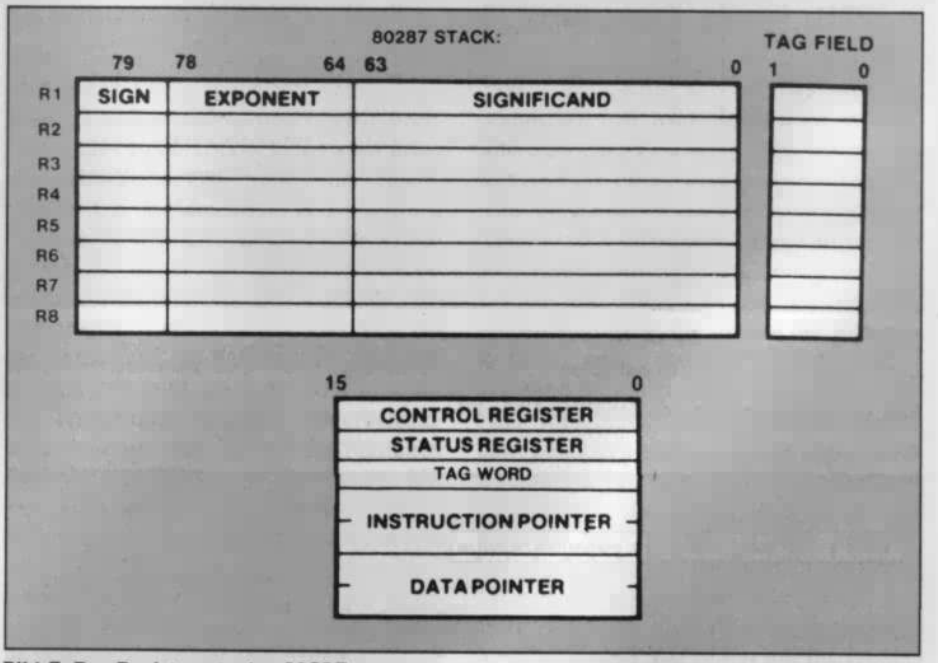


Bild 7. Der Registersatz des 80287

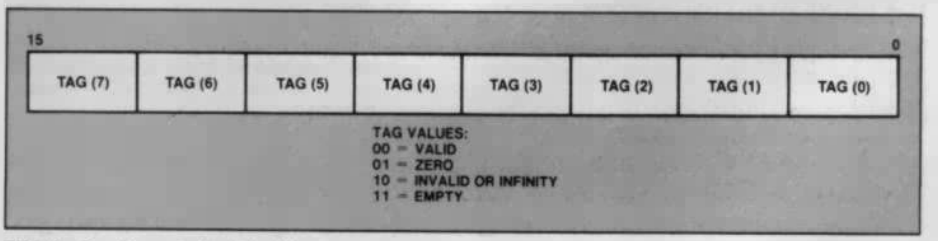


Bild 8. Das Format des Tag-Wortes

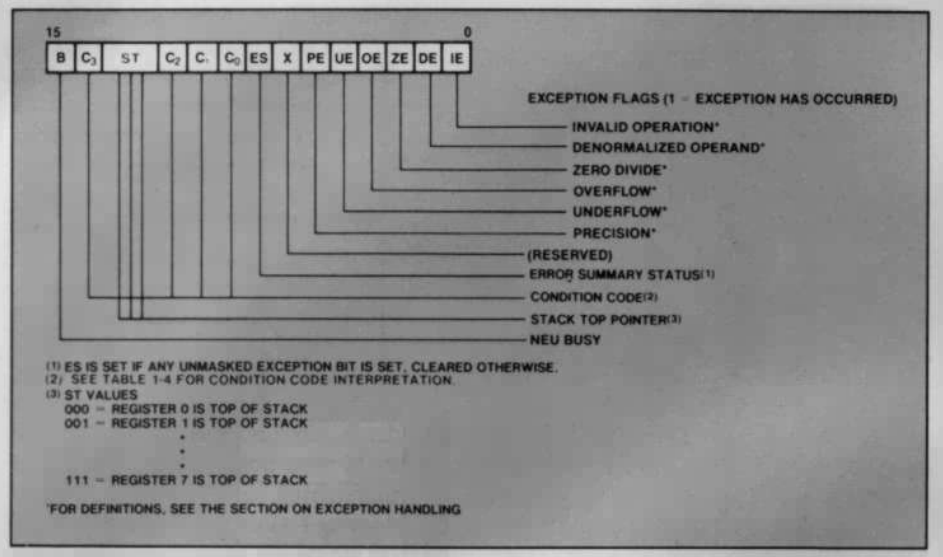


Bild 9. So ist das Status-Wort aufgebaut

Instruction Type	C_3	C_2	C_1	C_0	Interpretation
Compare, Test	0	0	X	0	ST > Source or 0 (FTST)
	0	0	X	1	ST < Source or 0 (FTST)
	1	0	X	0	ST = Source or 0 (FTST)
	1	1	X	1	ST is not comparable
Remainder	Q_1	0	Q_0	Q_2	Complete reduction with three low bits of quotient in C_0, C_3, and C_1
	U	1	U	U	Incomplete Reduction
Examine	0	0	0	0	Valid, positive unnormalized
	0	0	0	1	Invalid, positive, exponent = 0
	0	0	1	0	Valid, negative, unnormalized
	0	0	1	1	Invalid, negative, exponent = 0
	0	1	0	0	Valid, positive, normalized
	0	1	0	1	Infinity, positive
	0	1	1	0	Valid, negative, normalized
	0	1	1	1	Infinity, negative
	1	0	0	0	Zero, positive
	1	0	0	1	Empty Register
	1	0	1	0	Zero, negative
	1	0	1	1	Empty Register
	1	1	0	0	Invalid, positive, exponent = 0
	1	1	0	1	Empty Register
	1	1	1	0	Invalid, negative, exponent = 0
	1	1	1	1	Empty Register

NOTES:
1. ST = Top of stack
2. X = value is not affected by instruction
3. U = value is undefined following instruction
4. Q_n = Quotient bit n following complete reduction ($C_2 = 0$)

Bild 10. Die Bedeutung der Condition-Code-Bit

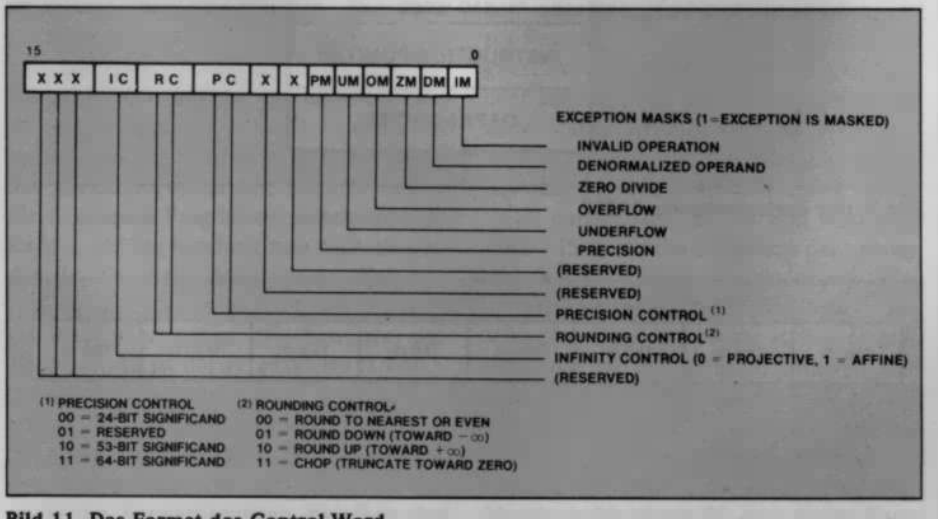


Bild 11. Das Format des Control Word

sich in 64 Mantissen- und 15 Exponenten-Bit sowie einem Vorzeichenbit auf. Jedem Register zugeordnet sind zwei Bit im sogenannten TAG FIELD. Ein „Tag" ist hier eine Identifikation über den Inhalt des zugehörigen Registers; *Bild 8* zeigt die Struktur aller Tags und deren Bedeutung im TAG WORD. Danach indizieren die Tags den Inhalt des zugehörigen Registers als GÜLTIG (VALID), NULL (ZERO), UNGÜLTIG/UNENDLICH (INVALID OR INFINITY) oder als LEER (EMPTY). Neben dem TAG WORD existieren noch das CONTROL- und STATUS-Register sowie der INSTRUCTION und DATA POINTER. Das *Bild 9* zeigt den Aufbau des STATUS-Registers. BUSY und der gleichnamige Pin zeigen den Arbeitsstatus der NEU an; die Bit C0...C3 sind mit den ähnlichen Flags der CPU vergleichbar. Wie die Tabelle in *Bild 10* zeigt, reflektieren sie den Zustand nach der Durchführung einer arithmetischen Operation. Bit 14, 13 und 12 stellen den Stackpointer ST dar, mit dessen Hilfe die gewünschten Operanden auf dem Registerstack adressiert werden. Der aktuelle Wert des ST beschreibt das oberste Stackelement, den „Top of Stack TOS". Viele numerische Instruktionen gestatten eine implizite Adressierung und beziehen sich dann auf den TOS-Operanden. Bei der expliziten Adressierung kann, ausgehend vom aktuellen ST-Wert, der i-te Operand im Registerstack angesprochen werden. Aufgrund dieser Organsiation kann man durch die relative Adressierung die modulare (Unterprogramm-)Programmierung vereinfachen, in dem man den Stack für die Parameter-Übergabe nutzt. Unabhängig vom aktuellen ST-Wert kann man die einzelnen Operanden durch ihren Abstand zum TOS adressieren! Die restlichen Bit des STATUS-Registers reflektieren den Fehler-Zustand nach numerischen Operationen. *Bild 11* zeigt den Aufbau und die Bedeutung der einzelnen Bit des CONTROL-Registers, mit dessen Hilfe der Programmierer die Arbeits-Modi des Numerik-Prozessors einstellen kann. Das niederwertige Byte enthält die Maskierungs-Bit zur Freigabe bzw. Sperrung einzelner Exceptions. Im höherwertigen Byte werden die Steuerung der Genauigkeit (Precision), der Rundung (Rounding) sowie der Unendlichkeit (Infinity) eingestellt. Nach dem Standard 745 arbeitet der NPX üblicherweise mit einer 64-Bit-Mantisse, deren Länge durch Variation der PC-Bits im Control Word jedoch auch auf Standards früherer Numerik-Prozessoren angepaßt werden kann. Neben der im Standard 745 geforderten Rundung (Round to nearest) gestatten die RC-Bits im Control Register die Einstellung weiterer Modi: Die Auf- bzw. Abrundung eines Wertes zum

nächstmöglichen, darstellbaren Wert in Richtung + unendlich bzw. – unendlich sowie das Abschneiden (Chopping) einer Zahl. Im letzteren Modus setzt der NPX den Nachkommateil einer Zahl auf Null und erzeugt so die nächste Ganzzahl (Integer) in Richtung 0.0! Mit Hilfe des „IC"-Bit stellt man die Behandlungsweise der Unendlichen ein. *Bild 12* zeigt in einer Gegenüberstellung beide Möglichkeiten. Aus der Mathematik bekannt und geläufig ist die affine Betrachtung, bei der die Zahlengerade links und rechts mit – unendlich und + unendlich als zwei völlig von einander getrennten Werten abgeschlossen wird. Die projektive Betrachtung verschmilzt beide zu einem einzigen Wert unendlich, auf den kreisförmig die negativen und positiven Zahlen zulaufen. Die rechnerinterne Darstellung nutzt diese mathematischen Modelle, wie an einem einfachen Beispiel gezeigt werden kann. Unter der Annahme, ein Byte diene zur Darstellung vorzeichenbehafteter Ganzzahlen, so gilt für den darstellbaren Wertebereich die Tabelle aus *Bild 13.* Inkrementiert man nun den größten positiven Wert, also „+ unendlich", so erhält man – unendlich. Umgekehrt wird dann aus – unendlich nach Dekrementieren + unendlich. Bietet die affine Darstellung auf den ersten Blick auch mehr Informationen, so birgt sie doch ihre Gefahren. Beispielsweise berechnet der NPX ein resultierendes Vorzeichen durch Verknüpfung der beiden Operanden-Vorzeichen. Aus diesem Grunde kann nach einer Berechnung ein X = +0.0 oder aber auch ein X = –0.0 auftreten. Für sich allein gesehen ohne Belang

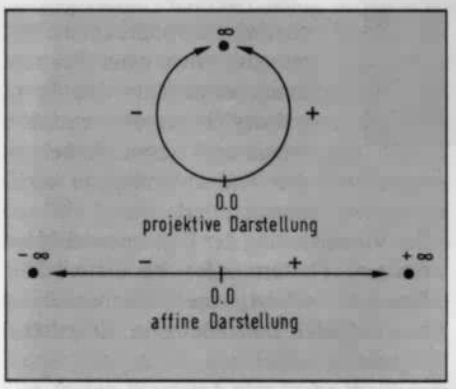


Bild 12. Projektiver und affiner Abschluß der reellen Zahlen

–∞				+∞
-128_{10}	≤	Byte	≤	$+127_{10}$
80_H	≤	Byte	≤	$7F_H$

Bild 13. Die Maschinenzahlen, die ein Byte darstellen kann, wenn das Vorzeichen berücksichtigt werden muß

Datentyp	Bits	Sign. Dezimale	Ungefährer Dezimalbereich
WORD INTEGER	16	4	$-32768 \le X \le +32767$
SHORT INTEGER	32	9	$-2*10^9 \le X \le +2*10^9$
LONG INTEGER	64	18	$-9*10^{18} \le X \le +9*10^{18}$
PACKED DECIMAL	80	18	$-99...99 \le X \le +99...99$
SHORT REAL	32	6-7	$8.43*10^{-37} \le \|X\| \le 3.37*10^{38}$
LONG REAL	64	15-16	$4.19*10^{-307} \le \|X\| \le 1.67*10^{308}$
TEMPORARY REAL	80	19	$3.4*10^{-4932} \le \|X\| \le 1.2*10^{4932}$

Bild 14. Die Datentypen des Bausteins 80287

Instruktions-Klasse	Typ
Daten-Transfer	*Daten laden, speichern und austauschen;*
Arithmetik	*Addition, Subtraktion, Multiplikation, Division; Umgekehrte Subtraktion und Division; Wurzel, Skalierung, Modulo, Integer-Teil bestimmen, Vorzeichen ändern, Absolut-Wert bestimmen, EXTRACT*
Vergleiche	*Daten vergleichen, untersuchen, testen;*
Transzendente	*Tangens, Arcustangens, $2^X - 1$, $Y*\log_2(X+1)$, $Y*\log_2(X)$*
Konstante	*0, 1, π, $\log_{10}(2)$, $\log_e(2)$, $\log_2(10)$, $\log_2(e)$*
Prozessor Steuer.	*CONTROL-Wort laden und speichern; STATUS-Wort speichern; NPX-Umgebung laden und speichern; Exceptions speichern, laden und rücksetzen; Initialisieren; PROTECTED MODE aktivieren;*

Bild 15. Die Befehlsklassen des 80287 und die eingebauten Konstanten

wird der NPX bei einer eventuell folgenden Division 1/X + oder – unendlich berechnen, in der affinen Betrachtung zwei völlig unterschiedliche Ergebnisse einer Operation für ein und denselben Ausgangswert! Nach einem Reset des Chips ist der projektive Modus eingestellt. [2] enthält tiefergehende Detail-Informationen, die für Systemprogrammierer nützlich sind. Wie bei der Vorstellung des Befehlssatzes noch näher erläutert wird, stellt der NPX u. a. für die Entwicklung eigener Fehlerbehandlungs-Routinen die beiden Instruktionen FSTENV und FSAVE zur Verfügung. Mit ihrer Hilfe erhält man eine komplette Übersicht der internen Register-Inhalte, indem der NPX diese ab der im Instruktions-Operanden übergebenen Adresse in das externe RAM transferiert. In diesem Zusammenhang wichtig sind der Instruction- und der Data-Pointer, die auf die zuletzt ausgeführte NPX-Instruktion verweisen.

Datentypen und Instruktionen.

Die Tabelle aus *Bild 14* zeigt die vom 80287 unterstützten numerischen Datentypen, deren Bitlänge bei interner Darstellung, sowie die Anzahl signifikanter Dezimalstellen in den jeweiligen Dezimalbereichen. Die Tabelle aus *Bild 15* zeigt die grundsätzlichen Instruktionen des 80287. Allgemein läßt sich sagen, daß der Instruktionssatz bis auf einige wenige Ausnahmen zum Vorgänger 8087 kompatibel ist. Die typische NPX-Instruktion nimmt einen oder zwei Operanden entgegen, führt ihre Tätigkeit aus und gibt ein Resultat zurück. Operanden sind dabei die Inhalte der internen Register oder externer Speicherplätze. Während für einige Instruktionen der Operand vordefiniert ist (implizit!), erlauben bzw. benötigen andere explizit angegebene Operanden. Im ersten Fall ist der Operand üblicherweise das oberste Stackelement, im letzteren ein beliebiger anderer Register- oder Speicherinhalt. Die beiden möglichen Operanden werden mit Quelle (src) und Ziel (dst) bezeichnet. Ein Quell-Operand wird in keinem Fall durch die Instruktion geändert, selbst wenn dieser zuvor in ein anderes Zahlenformat zu wandeln ist; eine solche Wandlung wird dazu in einem Prozessor-internen Arbeitsbereich durchgeführt! Im Gegensatz dazu kann ein Ziel-Operand durch die Instruktion verändert werden, wenn er mit deren Resultat überschrieben wird. Ein Großteil der Instruktionen erlaubt eine optionale Operanden-Angabe und kann dadurch auf mehrere Arten codiert werden.

Wird fortgesetzt

Dr. Klaus Dieter Brinkmann, Dr. Hans-Werner Eisermann

Aufbereitung von Bildern

Quantisierung, algorithmische Bearbeitung und Visualisierung von bildhaften Daten (Teil 1)

Bildhafte Daten sind durch Manipulation besser auswertbar. Es können zum Beispiel der Kontrast erhöht, Details erkennbarer gemacht und gestörte Informationen zurückgewonnen werden. Unterschiedliche Techniken bei der Visualisierung bestimmen den Eindruck beim Betrachter.

Mit der Verbindung von Europa und den Vereinigten Staaten von Amerika durch transatlantische Unterwasserkabel zu Beginn der 20er Jahre dieses Jahrhunderts begann ein reger Nachrichtenfluß zwischen beiden Kontinenten. Nach intensivem Austausch von Texten und Zahlenmaterialien kam erstmalig die Überlegung auf, Bildvorlagen in Zahlentabellen zu transformieren und über Kabel zu übertragen. Aus diesen erfolgreichen Anfängen der elektronischen Verarbeitung und Übertragung von bildhaften Vorlagen entwickelten sich alsbald diverse Spezialdisziplinen der Elektrotechnik und Informationsverarbeitung.

Mit der schnellen Innovation in der Halbleitertechnologie zu Mitte dieses Jahrhunderts und der damit einhergehenden Entwicklung von Großrechenanlagen konnte sich die Fachdisziplin der Digitalen Bildverarbeitung als selbständiger Zweig im Rahmen der Informations- und Nachrichtenverarbeitung etablieren. Eine rasche Verbreitung dieser Fachdisziplin erfolgte durch Problemlösungen in den Gebieten Astronomie und Medizin. Mit Bereitstellung der „Luftaufnahme-Satelliten" LANDSAT erschloß die Raumfahrt digitale Bilddaten einer breiten Anwendung in den zivilen Bereichen von Bodenschatzsuche, Ernte-Überwachung, Raumplanung und Umweltschutz.

Die revolutionsartigen Technologiesprünge der letzten Jahre ermöglichen die Bereitstellung von höchster Rechenleistung und großen schnell adressierbaren Datenspeichern durch relativ preiswerte Personal-Computer. Erst durch diese Entwicklung kann die volle Leistungsfähigkeit der Digitalen Bildverarbeitung heute einem breiten Anwenderkreis zur Verfügung gestellt werden, und es können neue Einsatzgebiete in der mittelständischen Industrie erschlossen werden.

Ziel und Anspruch der vorliegenden Arbeit ist eine Einführung und Erklärung der Thematik und Aufgabenstellung der Digitalen Bildverarbeitung. Benutzern von Personal Computern und Lesern ohne Informatik-Studium sollen an Hand von zahlreichen Beispielen mit Bildern und portablen Computer-Programmen vielfältige Verfahren dieses Gebietes erläutert werden.

Alle Programme dieses Beitrags sind in der Sprache FORTRAN geschrieben und auf Rechnern vom Typ Z240 advanced PC der Firma Zenith entwickelt, getestet und ausgeführt worden. Dieser IBM-kompatible PC besitzt neben der leistungsfähigen CPU 80286 einen mathematischen Coprozessor 80287 und wird unter MS-DOS 3.1 mit 8 oder 10 MHz betrieben.

Wenn nicht ausdrücklich anderes vermerkt ist, werden alle Beispiele mit Schwarzweißbildern der Größe 256 · 256 Bildpunkte und 256 Helligkeitsstufen berechnet. Zur Beurteilung des Zeitverhaltens ist anzumerken, daß die Rechenzeiten quadratisch zur Bildgröße sind, d. h. bei Verwendung von 128 · 128-Bildern sind die Rechenzeiten um den Faktor 4 geringer.

Zur Bearbeitung von Problemen aus dem Bereich der Digitalen Bildverarbeitung ist eine Konfiguration gemäß *Bild 1* erforderlich, um die folgenden Teilaufgaben zu erledigen.

- Durch Digitalisierung werden bildhafte Daten in den Rechner gelesen und nach Quantisierung abgespeichert.
- durch verschiedene Algorithmen werden diese gespeicherten Daten direkt bearbeitet oder nach einer Fourier-Transformation in den Spektralbereich werden einzelne Frequenzen gesondert behandelt und anschließend rücktransformiert.
- Zur Visualisierung der Ergebnisse auf Monitoren, Plottern oder herkömmlichen Druckern erfolgt eine Signalwandlung von digitaler Darstellung in elektrische Signale.

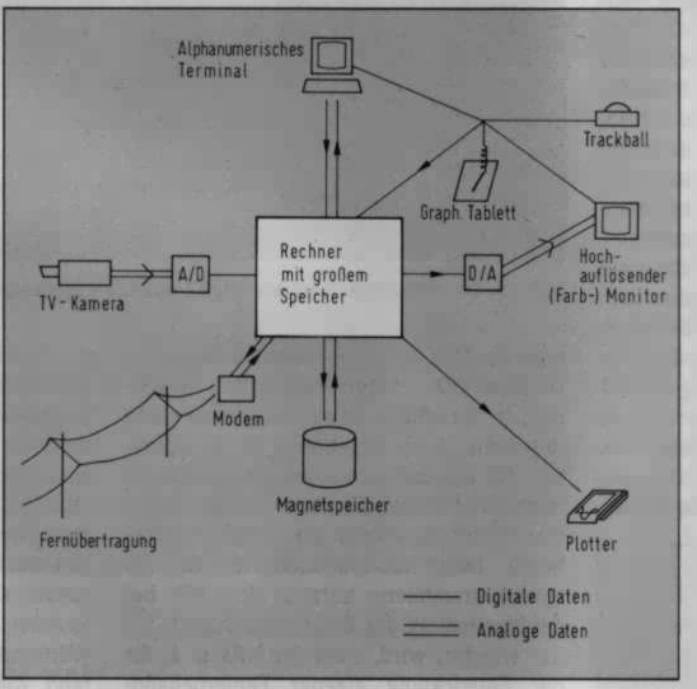


Bild 1. Maximal-Hardware für Bildverarbeitung

Bildaufnahme, Auflösung und Digitalisierung

Zum Einlesen von bildhaften Vorlagen in den Rechner ist eine Wandlung von visueller Information in durch Digitalrechner zu verarbeitende Information, d. h. in Zahlen, erforderlich. Der prinzipielle Ablauf dieser Informationswandlung ist in *Bild 2* dargestellt.

Mit Hilfe optoelektronischer Wandler wird die optische Information in elektrische In-

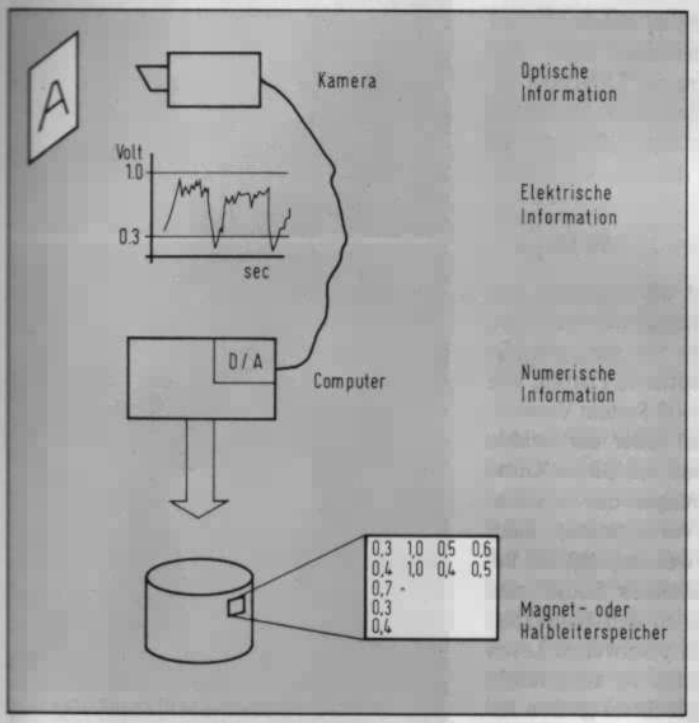


Bild 2. Die Digitalisierung von Bildern

formation umgesetzt, wobei der Fernsehnorm entsprechend der Signalpegel zwischen 0,3 V für Schwarz und 1,0 V für Weiß schwankt. Durch Analog-Digital-Wandlung werden aus den elektrischen Signalen Zahlenwerte, die direkt in den Rechner einzulesen sind, d. h. die elektrische Information wird in numerische Information gewandelt.

Die Wandlung der optischen Information in elektrische Signale erfolgt durch Helligkeitsmessung in der Vorlage und Erzeugung eines der Helligkeit proportionalen elektrischen Signals, wie z. B. Ladung, Strom oder Spannung. Üblicherweise wird hierfür gemäß *Bild 3* ein rechteckiges Raster benutzt. In jedem Feld dieses Rasters wird die durchschnittliche Helligkeit gemessen und in ein elektrisches Signal gewandelt, dessen Größe in einer diesem Raster entsprechenden Bild-Matrix als numerische Information zu speichern ist.

Normiert man die numerische Information auf den Wertebereich 1–256 (bzw. rechnerintern bei 8 Bit auf 0–255) und ordnet man z. B. der Farbe Schwarz in Bild 3a den Wert 256, der Farbe Weiß den Wert 1 und den dazwischen liegenden Grautönen die entsprechenden Werte zu, so werden für die vollständig schwarz gefärbten Rasterfelder (D1, E1) die Werte 256 gemessen.

In dem Raster F4 von Bild 3a wird eine der Farbe Grau entsprechende durchschnittliche Helligkeit von ca. 128 gemessen.

Bei der Visualisierung in Bild 3d ist zu erkennen, daß die scharfen Kanten der Vorlage „ausfransen“ und die Rasterstruktur klar erkennbar wird. Bedingt durch die Adreßarithmetik digitaler Rechner werden üblicherweise Raster von 64 · 64, 128 · 128, 256 · 256 oder 512 · 512 benutzt. Die von bekannten zivilen Aufklärungssatelliten gelieferten Bilder weisen wesentlich feinere Rasterungen auf. So liefert z. B. der amerikanische LANDSAT-Satellit je Geländeausschnitt 4 Bilder mit 2340 · 3234 Bildpunkten bei 128 Grauwertstufen. (Die Auflösung militärischer Satelliten liegt noch wesentlich höher.)

In der Regel sind die einzelnen Bildelemente quadratisch. Die Elemente des Rasters werden Bildpunkte genannt und oft mit dem englischen Acronym Pel (picture element) oder Pixel bezeichnet.

Unter Auflösung versteht man die Zahl der Bildpunkte pro Zeile. In den hier benutzten FORTRAN-Programmen wird diese Größe durch den Parameter PICMAX mittels PARAMETER-Statement festgelegt.

In *Bild 4* wird am Beispiel eines Personenlichtbildes mit verschiedenen Helligkeitswerten der Einfluß zwischen Auflösung und Detailerkennbarkeit dargestellt.

Vollständigkeitshalber soll hier noch erwähnt werden, daß wie in Bild 3a gezeigt bei konventioneller Fernsehtechnik die Abtastung des Helligkeitsrasters in der Art erfolgt, daß die Messung in der 1., 3., 5. Zeile zur Bestimmung des ersten Halbbildes dient und daß aus den Messungen der 2., 4., 6. Zeile das zweite Halbbild erstellt wird. In der Fernsehtechnik werden diese Halbbilder nacheinander dargestellt, während bei der Digitalen Bildverarbeitung bei-

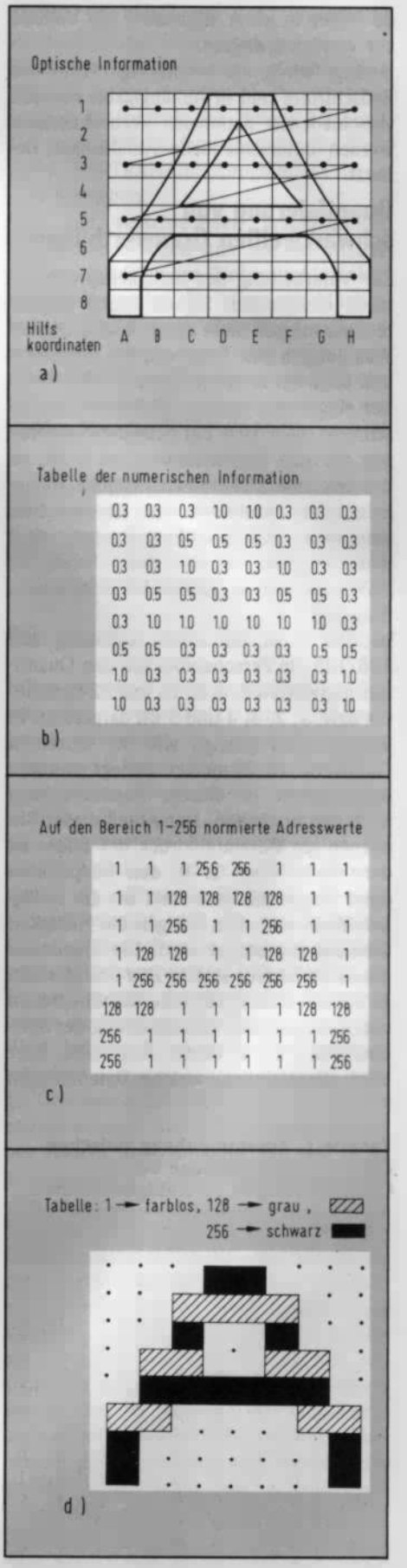


Bild 3. Digitalisierung, Rasterung und Quantisierung

de Bilder in einer Bildmatrix als Vollbild zur Verfügung stehen.
Bedingt durch die rechteckige Form des Bildschirmes und in Abhängigkeit von den verschiedenen nationalen Fernsehnormen werden unterschiedliche Auflösungen benutzt.

Quantisierung von schwarzweißen Grauwertbildern

Zur Verarbeitung der helligkeitsproportionalen elektrischen Signale durch Digitalrechner müssen diese durch Analog-Digital-Wandlung in eine Folge von Bits umgesetzt und über ein Interface direkt in den Rechner eingelesen werden. Üblicherweise beschränkt man sich bei Schwarzweißvorlagen auf eine Digitalisierung mit 8 Bit, so daß maximal 256 unterschiedliche Helligkeitsstufen abzubilden sind. In den hier benutzten FORTRAN-Programmen wird diese Größe durch den Parameter PGWMAX mittels PARAMETER-Statement festgelegt.
In *Bild 5* ist mit einer Auflösung von 128·128 ein Personenfoto mit den Quantisierungswerten 2, 4, 8, 16 und 32 Graustufen bzw. 1, 2, 3, 4 und 5 Bit dargestellt. In *Tabelle 1* ist gezeigt, wie die einzelnen Grauwerte in Bitmuster zerlegt werden. Man spricht in diesem Zusammenhang auch von Bitebenen. Der Anteil dieser Bitebenen am Gesamteindruck des Bildes ist darstellbar, wenn z. B. den Bildpunkten einer Ebene mit gesetztem Bit die Helligkeit Weiß und allen übrigen die Helligkeit Schwarz zugeordnet wird. Die Ergebnisse dieses Verfahrens sind in *Bild 6* dargestellt. In *Tabelle 2* ist für übliche Kombinationen von Auflösung und Quantisierung der Speicherbedarf in Kilobyte dargestellt. Hier wird offensichtlich, welche Datenmengen bei der Bildverarbeitung zu bearbeiten, zu speichern und zu übertragen sind. Wie schon erwähnt, übertragen LANDSAT-Satelliten je Geländeausschnitt vier Bilder mit 128 Graustufen und einer Auflösung von 2340 · 3234 Pixel, so daß sich die Datenmenge wie folgt ergibt:

2340 · 3234 · 7 · 4 bit = 25,259 MByte

Zum Vergleich sei auf die Kapazität von 20 MByte bei Festplattenlaufwerken und 1,2 MByte bei Disketten von gängigen PCs verwiesen. Neuartige optische Platten, wie sie z. B. im Optimem 1000 System Verwendung finden, haben auf jeder der beiden Seiten eine Kapazität von 1,2 GByte. Ohne zu sehr den Beschreibungen der verschiedenen Algorithmen vorzugreifen, kann schon hier festgestellt werden, daß zur Behandlung aller Punkte eines Bildes eine Übertragung des Bildes in den Hauptspeicher erforderlich ist, da pixelweises Lesen und Schreiben in Dateien zu zeitintensiv ist. Neben einem entsprechend großen Arbeitsspeicher ist somit insbesondere eine Software zur Verwaltung dieser Datenbereiche erforderlich.

Tabelle 1. Zusammenhang zwischen Grauwert und Bitebene bei Quantisierung in 16 Helligkeitsstufen

Grauwert	Bitebene 8	7	6	5	4	3	2	1
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	1	1
4	0	0	0	0	0	1	0	0
5	0	0	0	0	0	1	0	1
6	0	0	0	0	0	1	1	0
7	0	0	0	0	0	1	1	1
8	0	0	0	0	1	0	0	0
9	0	0	0	0	1	0	0	1
10	0	0	0	0	1	0	1	0
11	0	0	0	0	1	0	1	1
12	0	0	0	0	1	1	0	0
13	0	0	0	0	1	1	0	1
14	0	0	0	0	1	1	1	0
15	0	0	0	0	1	1	1	1

Speicherbedarf, subjektive Bildqualität und Farbbild-Verarbeitung

Von Huang wurden 1965 Experimente zur Messung des subjektiven Qualitätsempfindens bei unterschiedlichen Auflösungen und Quantisierungen durchgeführt. Hierzu wurde mehreren Personen ein Paßbild, die Aufnahme eines Kameramannes mit Himmel als Hintergrund und die Aufnahme von dicht besetzten Zuschauerrängen eines Sportstadions gezeigt. Von sämtlichen Fotos wurden den Betrachtern unterschiedliche Kombinationen von Auflösung und Quantisierung angeboten mit der Aufforderung, Bilder gleicher Qualität zu markieren. Die prinzipiellen Ergebnisse sind in *Bild 7* dargestellt und lassen erkennen, daß für Bilder mit geringem Detailreichtum Auflösung und Quantisierung in gewissen Bereichen austauschbar sind, während in sehr detailreichen Vorlagen eine geringe Auflösung nicht durch höhere Quantisierung ausgeglichen werden kann.
Wie aus der Farblehre bekannt und jedem wirksam am Beispiel des Farbfernsehens demonstriert, lassen sich weite Bereiche von Farbtönungen als Kombination aus drei Grundfarben Rot, Grün und Blau darstellen. Zur Digitalisierung von Farbvorla-

Bild 4. Auflösung und Detailerkennung

Bild 5. Quantisierung in unterschiedlicher Graustufenzahl

Tabelle 2. Speicherbedarf in KByte bei unterschiedlicher Auflösung und Quantisierung

Helligkeits-stufen	Auflösung 64	128	256	512
2	0.5	2	8	32
4	1	4	16	64
8	1.5	6	24	96
16	2	8	32	128
32	2.5	10	40	160
64	3	12	48	192
128	3.5	14	56	224
256	4	16	64	256

gen werden diese durch optische Filter in die drei Farbauszüge Rot, Grün und Blau zerlegt. Die Intensität dieser Farbauszüge wird dann entsprechend der Verarbeitung der Helligkeit in Schwarzweiß-Vorlagen behandelt. Im Vergleich zu Schwarzweiß-Bildern ist zur Speicherung von Farbbildern also das dreifache Volumen bei gleicher Auflösung und Quantisierung erforderlich.
Die wesentlichen der im folgenden zu beschreibenden Bildverarbeitungsalgorithmen lassen sich auch auf einzelne Farbauszüge anwenden.

Bildausgabe

Wesentlichen Einfluß auf die Wirtschaftlichkeit, Wirksamkeit und Akzeptanz von Methoden der digitalen Bildverarbeitung haben die Geschwindigkeit und Qualität der Visualisierung der Bilder. Ohne leistungsfähige Ausgabe auf Terminals und Hardcopy-Geräte wie Drucker, Plotter oder Filmschreiber wird diese neue Technologie keinen Einsatz finden.
Bedingt durch die gerätetechnische Vielfalt und die Dynamik der technologischen Entwicklung können hier nur einige Grundprinzipien erläutert und Standardausgabegeräte beschrieben werden.
Die einfachste und am schnellsten zu realisierende Methode der Bildausgabe benutzt die unterschiedliche Schwärzung druckbarer Zeichen von Schnelldruckern zur Darstellung verschiedener Grauwertstufen. Zur Darstellung des niedrigsten von 0 verschiedenen Grauwertes wird meist das Zeichen . verwendet, während intensive Schwärzung durch M, $ oder # erzielt wird.
Zur Erhöhung des darstellbaren Grauwertebereiches empfiehlt sich der Mehrfach-Druck einzelner Druckzeichen oder das Überdrucken mit anderen flächigen Symbolen. Für diese Art der Darstellung werden die Grauwerte 0–7 durch einfachen Druck erstellt. Für Grauwerte 8–11 werden die Zeichen einmal überdruckt, während der Bereich von 12–15 durch zweifaches Überdrucken dargestellt wird. Als Druckzeichen werden die Symbole der *Tabelle 3* benutzt. Der prinzipielle Aufbau eines hierfür geeigneten FORTRAN-Unterprogrammes ist in *Bild 8* gegeben.
Von den verschiedenen graphikfähigen Druckertypen sollen hier nur die Matrixdrucker (Nadel- oder Tintenstrahldrucker) vorgestellt werden. Sie bieten die Möglichkeit der Ansteuerung einzelner Druckernadeln bzw. Düsen. Damit sind sie geeignet zur Ausgabe von Binärbildern, können also Bilder erzeugen, die nur aus schwarzen und weißen Punkten bestehen. In den meisten

Tabelle 3. Zuordnung der Grauwerte zu Symbolen

Grauwert	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Symbol 1		.	,	:	+	I	X	M	$	+	(	%	%	I	#	*
Symbol 2										–	/	X	I	V	%	O	W
Symbol 3													V	%	O	W

Fällen werden 8 übereinanderliegende Nadeln der Druckmatrix zum grafischen Druck verwendet. In einem Durchgang können so 8 Zeilen eines Binärbildes dargestellt werden. Die Daten werden byteweise (1 Byte = 8 bit) vom Drucker verarbeitet; das höchstwertige Bit steuert dabei die oberste Nadel an. Jedem gesetzten Bit entspricht ein schwarzer Bildpunkt. Er wird dadurch erzeugt, daß die entsprechende Nadel auf das schwarze Farbband gedrückt wird. Bei einem weißen Bildpunkt ist das Bit nicht gesetzt; der abgebildete Bildpunkt behält die Farbe des Papiers.
Wird farbiges Druckband verwendet und mehrfach gedruckt, können effektvoll farbige Grafiken auch mit einfachen Geräten erstellt werden. Durch Ansteuerung einzelner Nadeln in Verbindung mit fein abgestuftem Papier- und Druckkopftransport können einige wenige Grauwertabstufungen realisiert werden.
Gängige Verfahren hierzu sind Mehrfachdruck, leicht verschobener Mehrfachdruck sowie Änderung der Anschlagstärke und Verfahren mit Raster-Druckverfahren, die im nächsten Abschnitt ausführlich beschrieben werden. In den meisten Fällen können diese Betriebsmodi direkt am Drukker eingestellt werden. Es ist dann zu beachten, daß der Zeilenabstand passend eingestellt wird, damit ein geschlossenes Druckbild entsteht. In *Bild 9* sind zwei Binärbilder zu sehen.
Zum Betrieb von Tintenstrahldruckern ist anzumerken, daß die Düsen bei nichtdurchgängigem (Dauer-)Betrieb des Druk-

Bitebene 4

Bitebene 3

Bitebene 2

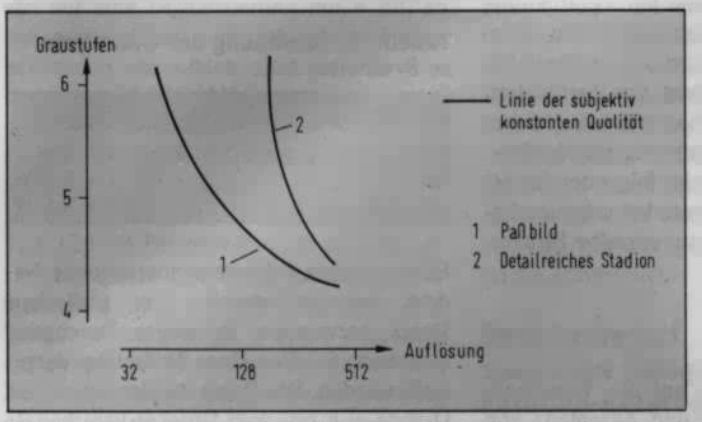
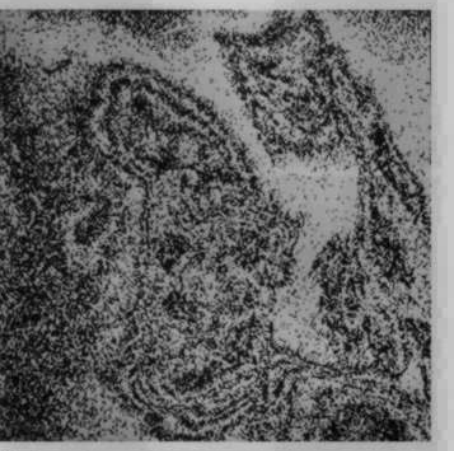


Bild 7. Subjektive Qualität, Auflösung und Quantisierung (prinzipiell nach Huang)

Bild 6. Einfluß der Bitebene auf den Bildein



Bitebene 1

```
        SUBROUTINE DRUCK (OUT,LPT)
C
      INTEGER  PICMAX
      PARAMETER  (PICMAX=256)
C
      INTEGER*2 OUT(PICMAX,PICMAX)
      LOGICAL DRUCK2,DRUCK3
      DIMENSION IAUS(3,128), ISYM(16,3)
      DATA ISYM/1H ,1H.,1H,,1H:,1H+,1HI,1HX,1HM,1H$,1H+,1H(,1H%,1H%,1HI,
     11H*,1H*,8*1H ,1H-,1H/,1HX,1HI,1HV,1H%,1HO,1HW,12*1H ,1HV,1H%,1HO,1
     2HW/
C
      DO 30 MZ=1,128,2
      DRUCK2=.FALSE.
      DRUCK3=.FALSE.
      DO 20 MS=1,128
C
C     ZUSAMMENFASSEN VON 2 ZEILEN WEGEN VERZERRUNG
C
      IGW=((OUT(MZ,MS)+OUT(MZ+1,MS))/32)+1
      IF (IGW.GT.16) IGW=16
      IF (IGW.GT.8) DRUCK2=.TRUE.
      IF (IGW.GT.12) DRUCK3=.TRUE.
      DO 10 MSY=1,3
   10 IAUS(MSY,MS)=ISYM(IGW,MSY)
   20 CONTINUE
      WRITE (LPT,5000)(IAUS(1,LX),LX=1,128)
      IF (DRUCK2) WRITE (LPT,5010)(IAUS(2,LX),LX=1,128)
      IF (DRUCK3) WRITE (LPT,5010)(IAUS(3,LX),LX=1,128)
   30 CONTINUE
      RETURN
 5000 FORMAT (1X,128A1)
 5010 FORMAT (1H+,128A1)
      END
```

Bild 8. Fortran-Programm für Schnelldrucker-Plot zur Ausgabe des Feldes OUT über die Unit LPT (ausgegeben wird ein 128 · 128-Bild)

kers leicht austrocknen. Sie bedürfen deshalb besonderer Pflege. Brauchbare Ergebnisse für grafische Darstellungen werden nur bei Verwendung von Spezialpapier mit besonderen Saugeigenschaften erzielt.

Besitzen grafische Ausgabegeräte die Fähigkeit, hinreichend kleine Druckpunkte wie z. B. einzelne Nadeln eines Nadeldruckers zu adressieren und getrennt schwarz oder weiß zu drucken, so lassen sich unterschiedliche Grauwerte mit Hilfe von sogenannten Druckmatrizen realisieren. Der Grauwert eines einzelnen Bildpunktes wird hierbei durch eine Matrix von Druckpunkten realisiert, ähnlich wie bei der Darstellung von Bildern in Zeitungen. Die Zahl der schwarz zu druckenden Punkte einer so definierten Druckmatrix entspricht hierbei dem zu realisierenden Grauwert. Formt man beispielsweise aus 4 Punkten je Zeile und 4 Zeilen eine 16 Punkte umfassende Matrix, so wird der Grauwert 0 = Weiß durch die „leere Matrix" definiert, d. h. es wird keines der adressierbaren Felder geschwärzt.

Für die Grauwerte 1–16 muß jeweils eine entsprechende Anzahl von Feldern ge-

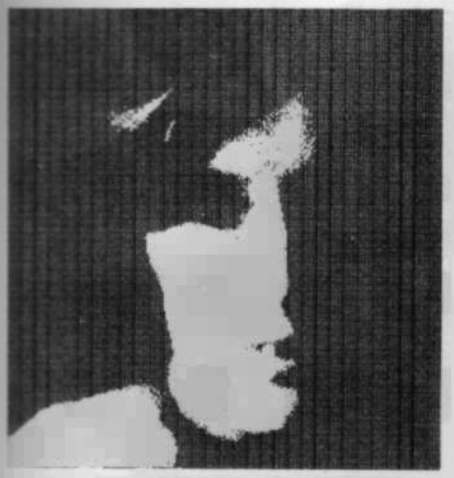


Bild 9. Bildausgabe auf Epson-Drucker

schwärzt werden. Entscheidend für die Bildqualität ist hierbei die Anordnung der zu schwärzenden Punkte. Drei unterschiedliche Methoden sollen beschrieben und am Beispiel eines Graukeiles und des Personenfotos von Bild 5 demonstriert werden. Allen Verfahren ist gemeinsam, daß durch eine N·N-Matrix insgesamt $N^2 + 1$ unterschiedliche Grauwerte dargestellt werden können. Darstellungen mit 10 Grauwerten (N = 3) liefern bereits Bilder akzeptabler Qualität.

Lineares Schwärzen der Druckmatrix

Werden die Felder einer 4·4-Matrix zeilenweise durchnumeriert, so erhält man einen eindimensionalen Vektor der Länge 16. Die lineare Schwärzung erfolgt derart, daß für Grauwert 1 das erste Element dieses Vektors geschwärzt wird, für Grauwert 2 werden das erste und das zweite Element geschwärzt usw. D. h. zur Darstellung des Grauwertes n werden die Felder 1 bis n geschwärzt. In *Bild 10* sind für diese Darstellungsart die entsprechenden Druckmatrizen, der Graukeil und das Personenfoto wiedergegeben. Wie am Graukeil deutlich zu sehen ist, entstehen insbesondere in Flächen gleichen Grauwertes störende geometrische Muster und Strukturen.

Dithermatrix

Zur Vermeidung der Ausbildung von störenden geometrischen Strukturen versucht man unter gerätespezifischen Gesichtspunkten sogenannte Dithermatrizen zu benutzen. Im Gegensatz zum vorherigen Verfahren wird hierbei für jeden einzelnen Grauwert n durch Belegung von nPunkten der Matrix ein vorgegebenes Muster bestimmt, das in der Regel nicht der linearen Schwärzung entspricht. Je nachdem, ob überwiegend flächige oder mehr linienhafte Bilder auszugeben sind, können unterschiedliche Dithermatrizen eingesetzt werden.
In *Bild 11* sind für die Werte 1–16 die 4·4-Dithermatrizen sowie der Graukeil und das Personenfoto dargestellt.
Die Ausbildung periodischer Strukturen ist im Vergleich zu Bild 10 bei nur unwesentlich erhöhtem Software-Aufwand wesentlich vermindert worden.

Bildung von Druckmatrizen mit Zufallsgenerator

Zur Vermeidung jeglicher Strukturen wird in *Bild 12* die Schwärzung der einzelnen Punkte der 4·4-Matrix durch einen Zufallsgenerator vorgenommen. Es entstehen hierdurch keine regelmäßigen Strukturen mehr, doch wirken die Bilder insgesamt unruhiger.
Da trotz fortschreitender Technik die meisten Ausgabegeräte nur Schwarzweiß-Druck ermöglichen, ist die zu beobachtende Qualitätserhöhung im wesentlichen durch eine weitere Vergrößerung der Auflösung erzielt worden. Zusätzliche Modifikation der Rastermethoden, wie z. B. die zufallsgesteuerte Addition von Korrekturtermen zur Verminderung des „Rastereffektes" tragen zu einer weiteren Verbesserung bei.

Ausgabe auf Standardterminal

Im Gegensatz zum Drucker erfolgt am Terminal durch Mehrfach-Druck keine Änderung der Grauwertintensität, da jeweils beim Drucken eines Zeichens das vorherige gelöscht wird. Unterschiedliche Grauwerte lassen sich also nur durch verschieden flächige Symbole erzeugen. Bei Benutzung des Standardzeichensatzes wird hierdurch die Vielfalt darstellbarer Grauwerte eingeengt. Weiterhin ist zu beachten, daß bei gängigen Terminals im wesentlichen 80 Zeichen je Zeile darstellbar sind und daß nur bis zu 30 Zeilen auf den Schirm passen. Erschwerend kommt noch hinzu, daß die so adressierbaren Bildpunkte im allgemeinen nicht quadratisch sind. Geringe Auflösung und Verzerrung behindern somit die Bildauswertung extrem stark. Eine gewisse Verbesserung des Umfanges der Grauwertdarstellung bietet die Umschaltung auf Grafikzeichen (z. B. MS-DOS-Befehl: GRAFTABL). Durch die Auswahl geeigneter flächiger Symbole kann erreicht werden, den im allgemeinen störenden Abstand zwischen den Zeilen und Zeichen so zu vermindern, daß Flächen gleicher Grauwerte keine Kanten und zeichentypischen Strukturen aufweisen, sondern gleichmäßig geschwärzt erscheinen.

Neuere Terminals wie das ZENITH 1240 E gestatten die Umschaltung zwischen zwei Helligkeitsstufen (normal, bold). Bei gleichzeitiger Verwendung von flächigen Grafiksonderzeichen lassen sich akzeptable Grauwertdarstellungen mit bis zu neun Helligkeitsstufen realisieren, wie das Beispiel eines Graukeils auf solch einem Terminal *in Bild 13* zeigt.

Nahezu als Standarderweiterung zu PCs wie dem IBM-kompatiblen ZENITH 240 advanced PC werden sogenannte Grafik-Karten angeboten, die einerseits die Auflösung des Bildschirmes wesentlich verbessern und andererseits den Grauwertumfang erhöhen und/oder Farbdarstellung ermöglichen. Als Standard für die Leistung dieser Zusatzkarten gilt mittlerweile die EGA-Norm. Wesentlich vorangetrieben wurde die Entwicklung dieser Grafik-Karten durch die Firma Hercules Computer Technology (HCT), so daß andere Anbieter nur mit „Hercules-kompatiblen" Karten auf dem Markt erfolgreich konkurrieren können.

Als Software für diese Karten kann z. B. GRAPH X von HCT benutzt werden. Dieses Produkt gestattet von Assembler, Basic und FORTRAN das Zeichnen von Punkten, Linien und Kreisen sowie das Ausfüllen von Rechtecken und die Darstellung von Texten. Hierdurch erhält man eine Auflösung von 720 Punkten je Zeile und 348 Zeilen je Bildschirm bei Monochrom-Betrieb. Bei Verwendung von 4·4-Druckmatrizen für 17 Grauwerte ergibt sich eine nutzbare Auflösung von 180 · 87 Bildpunkten. 3·3-Matrizen für 10 Grauwerte ergeben eine Auflösung von 240 · 116 Bildpunkten.

Beachtet werden muß jedoch, daß, wie bei Standardterminals, die Bildpunkte nicht quadratisch sind, sondern ein Seitenverhältnis von 2:3 aufweisen. Können die

Bild 10.

Bild 11.

Bild 12. Der Bildeindruck bei verschiedenen Druckmatrizen: lineares Schwärzen (Bild 10), mit Dithermatrizen (Bild 11) und Besetzung durch Zufallsgenerator (Bild 12); jeweils mit Graukeil, Personenfoto und Druckmatrizen

Bildpunkte zusätzlich in zwei Helligkeitsstufen ausgegeben werden, so ergeben sich bei 4·4-Druckmatrizen 33 mögliche Grauwertstufen und bei 3·3-Matrizen 19 Stufen. Die Qualität solcher Bilder ist aus *Bild 14* zu ersehen. In dem uns von Zenith Data Systems zur Verfügung gestellten System wird eine Grafik-Karte von Paradise benutzt.

Ausgabe auf grafisches Terminal

Mit dem Begriff „grafische Terminals" werden in dieser Arbeit solche Systeme bezeichnet, die es gestatten, jeden einzeln adressierbaren Punkt des Bildschirmes in mehr als zwei unterschiedlichen Helligkeiten darzustellen. In der Regel handelt es sich hierbei um Rasterdisplaysysteme. Ein meist ausbaubarer Bildspeicher ist so organisiert, daß in jeder Bitebene pro Bildpunkt des Schirmes ein Bit gespeichert werden kann. Durch Ausbau auf n Speicherebenen sind dann je Pixel 2^n Werte möglich, die in Umkehr des vorher beschriebenen Verfahrens hier digital-analog gewandelt und in Form von Videosignalen auf Fernsehmoni-

Bild 13. Grauwertdarstellung auf einem PC

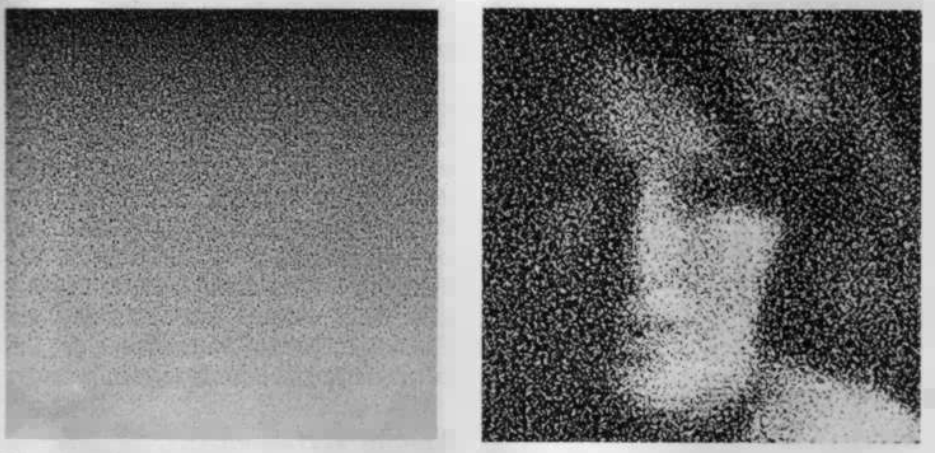


Bild 14. Zufallsbesetzung bei zweistufiger Helligkeit (links der Graukeil)

toren sichtbar gemacht werden. Um das Bearbeiten von Bildern zu beschleunigen, verwendet man in modernen Systemen die aus dem Bildspeicher gelesenen Daten nicht zur Erzeugung des Videosignals, sondern als Adresse für Daten einer Lookup-Tabelle. Die hiermit adressierten Daten werden in Videosignale gewandelt. Der Vorteil des Verfahrens ist am folgenden Beispiel erkennbar:

Soll ein 512·512-Bild mit 8-Bit-Grauwerten als Negativ dargestellt werden, so muß an allen 512 · 512 = 262 144 Speicherplätzen der Grauwert in die Negativdarstellung umgesetzt werden.
Bei Verwendung einer Lookup-Tabelle müssen lediglich die 2^8-Werte der Tabelle in die Negativdarstellung gewandelt werden. Der Aufwand zwischen beiden Verfahren ist 1024 : 1.

Werden durch die Daten aus dem Bildspeicher drei Lookup-Tabellen adressiert, so können deren Daten zur Erzeugung der Grundfarben Rot, Grün und Blau bei Farbwiedergabe nach dem RGB-Verfahren dienen. Moderne Bildverarbeitungssysteme haben in der Regel mehrere Lookup-Tabellen mit großer Wortlänge, um diese zum Rechnen zu benutzen.

Bildverarbeitung im Ortsbereich

Die eigentliche Aufgabe der Bildverarbeitung besteht darin, die Bilddaten so aufzubereiten, daß eine bessere Auswertung und Weiterverarbeitung möglich ist. In der Praxis wird der Begriff Bildverarbeitung meist weiter gefaßt und die Interpretation der Daten als Teilaufgabe betrachtet. Wir verstehen hier die Aufgaben der Bildverarbeitung beispielsweise so, daß LANDSAT-Aufnahmen in bezug auf Kontrasterhöhung und Detailerkennbarkeit behandelt werden. Die anschließende Zuordnung der fotografierten Bodenflächen zu Wald, Landwirtschaft, Industrie usw. ist für uns eine Aufgabe der Mustererkennung und wird hier nicht behandelt.
Bei der Bildverarbeitung ist zu unterscheiden, ob die im Bild vorhandene Information besser aufbereitet und dargestellt werden soll oder ob der Versuch unternommen wird, gestörte Information durch inverse Filterung zurückzugewinnen. Im ersten Fall spricht man von Bildverbesserung (image enhancement) während die zweite – anspruchsvollere – Aufgabe als Bildrestauration (image restauration) bezeichnet wird.
Unter Arbeiten im Orts- oder Bildbereich versteht man die direkte Behandlung der einzelnen Pixel eines Bildes. Dagegen werden die im Teil 2 (nächstes mc-Heft) beschriebenen Verfahren im Frequenzbereich angewendet. Hierzu ist eine Transformation des Bildes in Spektralkomponenten und Bearbeitung von „Wellen" unterschiedlicher Frequenz erforderlich (Fourier-Transformation).
Sowohl die Leistungsfähigkeit als auch die Verarbeitungsgeschwindigkeit hängen davon ab, ob alle Pixel eines Bildes nach gleicher Vorschrift zu bearbeiten sind oder ob die unterschiedlichen Einflüsse benachbarter Punkte zu beachten sind.
Man unterscheidet deshalb globale Verfahren, die auf alle Bildpunkte eines Bildes gleichartig wirken, und ortslokale Verfahren, die in Abhängigkeit von benachbarten Bildpunkten wirken. Weiterhin ist zu beachten, daß Verfahren linear oder nichtlinear wirken können. Wenn nicht besonders erwähnt, sind alle beschriebenen Verfahren von linearer Art.

Globale Verfahren

Da die globalen Verfahren die Pixel nur in Abhängigkeit des Grauwertes behandeln, bietet sich eine Realisierung dieser Verfahren durch Lookup-Tabellen-Technik an. Hierbei ist es unerheblich, ob diese Tabelle zur Steuerung von Bildausgabe-Geräten benutzt wird oder der internen Berechnung dient. In dem noch vorzustellenden Software-Paket PCBILD zur Bildverarbeitung auf PCs (Teil 2 im nächsten mc-Heft) wird eine Lookup-Tabelle durch einen Datenvektor VLT(256) vom Typ Integer realisiert. Sind keine Ausgabegeräte mit Lookup-Tabelle vorhanden, so kann z. B. durch den Befehl COPVLT (COPy Via Look-up-Table) das Bild in der Weise umgesetzt werden, daß die Grauwerte als Adresse in der Tabelle dienen. Dem Pixel werden dann die auf dieser Adresse stehenden Daten als Grauwert zugewiesen. Mit Ausnahme des Verfahrens der Linearen Spreizung kann keine Modifikation mit herkömmlicher (analoger)

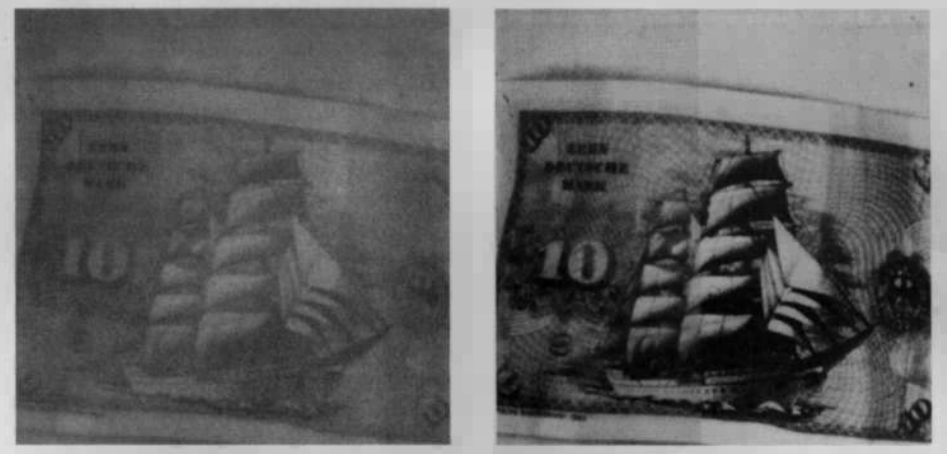


Bild 15. Gegenüberstellung der Kontrastverbesserung durch Spreizung (im linken Teil des rechten Bildes) und durch Histogramm-Linearisierung (im rechten Teil des rechten Bildes) im Vergleich zum Original (ganz links)

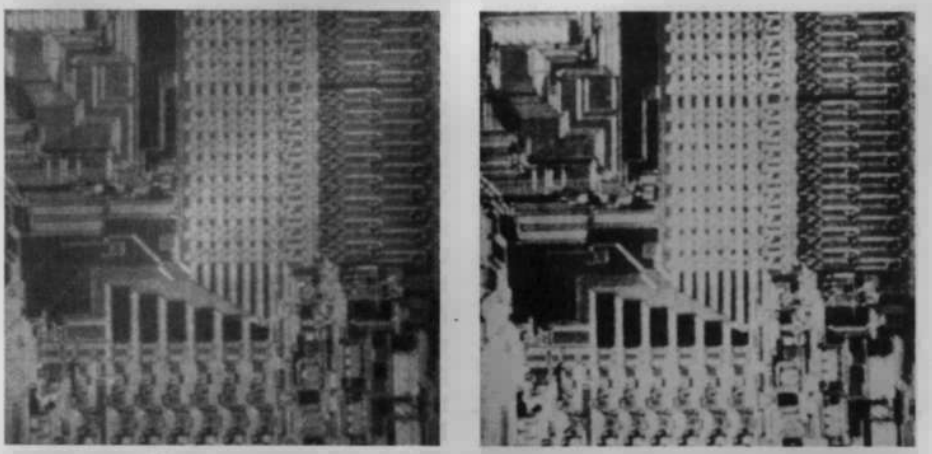


Bild 16. Einfluß der Histogramm-Linearisierung (rechts) auf die Aufnahme einer integrierten Schaltung (links)

Fotolabor-Technik nachvollzogen werden.
Zur vollen Ausnutzung aller möglichen darstellbaren Grauwerte wird die Lookup-Tabelle linear besetzt:

```
VLT(N) = A * N + B
```

Die Koeffizienten A und B werden bei 256 darstellbaren Grauwerten so gewählt, daß der minimale Grauwert IGWMIN im Bild auf 0 und der maximale Grauwert IGWMAX auf 255 transformiert wird.
Bezeichnet PGWMAX den maximal darstellbaren Grauwert, so ergibt sich in FORTRAN-Notation folgende Berechnungsvorschrift:

```
   PGWMAX = 256
   A      = FLOAT (PGWMAX - 1) /
            (IGWMAX - IGWMIN)
   B      = -A * IGWMIN
   DO 10 N = 1,PGWMAX
10 VLT(N) = A * N + B
```

Die lineare Spreizung der Kennlinie entspricht der fotografischen Praxis unterschiedlicher Entwicklungszeiten zur Kontrasterhöhung. Ergebnisse dieser Technik sind in *Bild 15* gezeigt.
Zur Bewertung der Grauwertverteilung im Bild wird eine Statistik der Häufigkeit der einzelnen Grauwerte aufgestellt, deren grafische Darstellung Histogramm genannt wird. Üblicherweise werden im Histogramm die relative Einzelhäufigkeit und die Summenhäufigkeit gemeinsam dargestellt. Zusatzinformationen wie Mittelwert und Entropie erleichtern die Bewertung des Bildes.
Mit Verfahren der Nachrichtentheorie kann gezeigt werden, daß durch Annäherung der Grauwertverteilung an vorgegebene Funktionsverläufe eine bessere Bildqualität zu erzielen ist. Hierzu werden Gruppen von verschiedenen Grauwerten zusammengefaßt und mit einem – möglicherweise völlig verschiedenartigen – neuen Grauwert dargestellt. Erreicht wird dieses dadurch, daß aus der gegebenen Statistik und der gewünschten Histogramm-Form eine neue Lookup-Tabelle berechnet wird. Größere Verbreitung haben im wesentlichen die Anpassungen an die folgenden Verteilungen gefunden

- Linear H ~ constant
- Hyperbolisch H ~ 1/Grauwert+constant
- Exponentiell H ~ exp (–Grauwert · const)

Zusätzlich wird oftmals die interaktive Vorgabe einer Verteilungsform mit Trackball, Joystick o. ä. benutzt.
Die Ergebnisse einer Histogramm-Linearisierung sind in Bild 15 denen der Linearen Spreizung gegenüber gestellt. In *Bild 16* ist eine Integrierte Schaltung gezeigt, die bereits so aufgenommen wurde, daß eine lineare Spreizung nicht mehr möglich ist. Durch die Histogramm-Linearisierung wird die Erkennung von Details wesentlich erleichtert, da die Aufnahme von größerer Schärfe und Brillianz zu sein scheint.
In Bilddarstellungssystemen mit Lookup-Tabelle können durch interaktive Modifikation der Lookup-Tabelle gezielt einzelne Grauwert-Bereiche hervorgehoben oder unterdrückt werden. Wird beispielsweise eine dreiecksförmige Lookup-Tabelle wie in *Bild 17* kontinuierlich im Bereich von 0–256 verschoben, so werden nacheinander die Bildteile mit Grauwerten in den Flankenbereichen dieses Dreiecks hell dargestellt. Es entsteht somit scheinbar ein Filtereffekt, der zur Verbesserung der Detailerkennbarkeit benutzt werden kann.

Lokale Verfahren mit Koeffizientenmatrix

Bei lokalen Verfahren werden zur Berechnung des Grauwertes IGNEU eines Pixels B(x,y) die Grauwerte in der Umgebung des Punktes untersucht. Bei dieser Gruppe von Filtern wird die Berechnung des Ergebnisses durch eine gewichtete Summierung der Grauwerte in der Umgebung des zu untersuchenden Bildpunktes B(x,y) vorgenommen.
Die Gewichtsfaktoren werden durch eine Koeffizientenmatrix C bestimmt, die so zu verstehen ist, daß der Mittelpunkt dieser Matrix – bei einer 3×3-Matrix also C(2,2) – den Koeffizienten für den Bildpunkt B(x,y) kennzeichnet. Die Wichtung der benachbarten 8 Punkte (B(x–1,y+1), B(x,y+1),B(x+1,y+1),B(x–1,y), B(x+1,y),B(x–1,y–1),B(x–1,y),B(x–1,y+1)) wird durch die übrigen Elemente der Matrix festgelegt.
Diese Matrix muß dann „über das Bild geschoben werden" und für jeden Bildpunkt ist Produktbildung und Summation über die Größe der Matrix erforderlich. Bei der Pro-

Bild 17. Wirkung interaktiv definierter Look-up-Tabellen auf die Original-Aufnahme des Geldscheins aus Bild 15

grammierung dieser Verfahren ist es unbedingt erforderlich, die unterschiedlichen Zugriffsmechanismen auf Elemente in mehrdimensionalen Feldern zu beachten, da sonst extrem lange Rechenzeiten die Folge sind.

Tiefpaß

Kennzeichnend für den Tiefpaß ist die gleichmäßige Wichtung aller Nachbarpunkte des Pixels B(x,y). Die Wirkung des Tiefpaßfilters wird beschrieben durch die Koeffizientenmatrix

$$C = \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Meist wird auf eine Normierung der Koeffizientenmatrix auf den Betrag 1 verzichtet und erst das Ergebnisbild mittels linearer Spreizung auf den darstellbaren Grauwertebereich normiert.
In FORTRAN-Notation erfolgt die Berechnung des Ergebnisbildes OUT aus dem Eingangsbild IN mit der Koeffizientenmatrix C der Größe MC×NC wie folgt:

```
      DO 200 IX = .....
      DO 200 IY = .....
      OUT(IY,IX) = 0
      NXX = NC/2+1
      MYY = MC/2+1
      DO 100 NX = 1,NC
      DO 100 MY = 1,MC
100   OUT(IY,IX) = OUT(IY,IX) +
      C(M,N)*IN(IY-MYY+MY,
      IX-NXX+NX)
200   CONTINUE
```

Die Wirkung des Filters besteht darin, daß geringe Grauwerte-Schwankungen herausgemittelt werden. Hierdurch kann Rauschen vermindert oder unterdrückt werden. Gestochen scharfe Bilder erscheinen nach der Filterung verwaschen. Dieser Effekt wird bei Vergrößerung der Umgebung verstärkt.

Laplace-Filter

Beschrieben wird der Laplace-Filter – auch Laplace-Operator genannt – durch die Koeffizientenmatrix

$$C = \begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

Die Benutzung dieser Matrix zur Berechnung wie beim Tiefpaß ist im Vergleich zur direkten Programmierung

OUT(IY,IX)=IN(IY–1,IX)+IN(IY+1,IX)
+IN(IY,IX–1)+IN(IY,IX+1)–4*IN(IX,IY)

zu langsam und sollte deshalb vermieden werden.
Die Wirkung des Laplace-Filters besteht in einer Verstärkung von Grauwertsprüngen. Ecken und Kanten werden durch diese Filterung hervorgehoben.

Roberts-Operator

Im Unterschied zu den vorherigen Filter-Operatoren wirkt der Roberts-Operator auf eine 2×2-Umgebung und ist durch 2 Koeffizientenmatrizen C0 und C1 bestimmt.

$$C0 = \begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix} \qquad C1 = \begin{matrix} -1 & 0 \\ 0 & 1 \end{matrix}$$

Die Ergebnisse der gewichteten Additionen mit C0 und C1 werden betragsmäßig addiert. Die Wirkung ist ähnlich der des Laplace-Filters. Auch hier ist die direkte Programmierung dem Arbeiten mit Matrizen vorzuziehen.

Sobel-Operator

Wie der Roberts-Operator wird auch der Sobel-Operator durch 2 Matrizen C0 und C1 beschrieben. Die Ergebnisse der Wichtung werden betragsmäßig addiert.

$$C0 = \begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix} \qquad C1 = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

In *Bild 18* ist die Wirkung des Sobel-Operators zur Erkennung von Kanten gezeigt.

Frei programmierbare Filterkoeffizienten

Aus der Literatur sind vielfältige Erweiterungen der beschriebenen Filter auf unterschiedliche Formen und Anzahl von Koeffizientenmatrizen bekannt. Grundprinzip ist hierbei, daß für jeweils besondere Vorzugsrichtungen oder Strukturelemente im Bild optimale Matrizen definiert werden. Wie im *Bild 19* zu sehen ist, können mit der Matrix

$$C = \begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix}$$

besonders gut senkrechte Kanten und Linien gefunden werden. Zur Erkennung von Kanten mit einer Neigung von 45 Grad ist, wie in *Bild 20* gezeigt, die Matrix

$$C = \begin{matrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & -1 \\ 1 & 1 & 0 & -1 & -1 \\ 1 & 0 & -1 & -1 & -1 \\ 0 & -1 & -1 & -1 & -1 \end{matrix}$$

gut geeignet. Je nach Anwendung entwirft man für spezielle Strukturen geeignete Filtermatrizen und kombiniert deren Ergebnisse miteinander. Im System PCBILD stehen hierfür Befehle zur Definition von beliebigen Koeffizientenmatrizen bis zu einer Größe 15 · 15 bereit. Durch Befehle zur Addition, Subtraktion, Multiplikation und Division von Bildern können die Resultate der verschiedenartigen Filterungen fast beliebig kombiniert werden.
Praktische Erfahrungen zeigen, daß nur in Spezialfällen der Einsatz von Filtermatrizen größer als 5 · 5 erforderlich ist. Hierbei ist im wesentlichen zwischen Rechenzeit und Leistung abzuwägen. *Tabelle 4* zeigt den Einfluß der Filtermatrizen auf die Rechenzeit.

Tabelle 4. Die Rechenzeiten des Bildverarbeitungssystems PCBILD auf dem Zenith PC 240 für INTEGER*2-Bilder der Abmessung 256·256 unter verschiedenen Filtermatrizen

Filtertyp/Operation (Minimum/Maximum Bestimmung)		Rechenzeit in s
Laplace direkt	3·3	61
Laplace mit Matrix	3·3	252
Roberts-Operator	2·2	54
Sobel-Operator	3·3	142
Programmierbare Matrix	5·5	400
Programmierbare Matrix	7·7	735
Addition	Bild 1 + Bild 2	40
Subtraktion	Bild 1 – Bild 2	40
Multiplikation	Bild 1 · Bild 2	40
Division	Bild 1 / Bild 2	60

Lokale Verfahren ohne Koeffizientenmatrix

Im Gegensatz zu den im vorherigen Abschnitt beschriebenen Verfahren benutzen die beiden folgenden Verfahren keine Matrizen, sondern nehmen aus den Daten der Umgebung eine Berechnung des Ausgangsbildes vor.

Mit Rangordnungsfiltern beschreibt man Verfahren, die aus der Ordnung der Grauwerte ein Ergebnis bestimmen, indem die Werte in aufsteigender Folge sortiert werden und auf definierte Positionen dieser Folge zugegriffen wird.

Tabelle 5 illustriert an einem Beispiel die Vorgehensweise für einen Rangordnungsfilter der Größe 3 · 3.

Tabelle 5. Die Vorgehensweise bei einem 3·3-Rangordnungsfilter an einem Beispiel

22	18	7	1	2	3	4	5	6	7	8	9	Position
4	2	5	Minimum							Maximum		
11	3	6	2	3	4	5	6	7	11	18	22	
Umgebung			Folge der geordneten Grauwerte									

Der frei wählbare Rang n des Filters gibt die Position in der Folge an, auf der das Ergebnis steht. Für Rang 7 ist der 7. Wert der Folge, d. h. im Beispiel 11 zu nehmen.

Ein Rangordnungsfilter vom Rang 9 liefert entsprechend 22. Einen Sonderfall stellt der Medianfilter dar. Hier wird immer der an mittlerer Position der Folge stehende Wert genommen. Im Beispiel der auf Position 5 stehende Wert 6.

Je nach Größe und Rang n werden Filter dieser Art zur Kantendetektion oder Tiefpaßfilterung eingesetzt. Trotz ähnlicher Wirkung handelt es sich jedoch im Gegensatz zu Tiefpaß- und Laplace-Filter um völlig verschiedenartige Filter, die zum Beispiel nicht durch eine Übertragungsfunktion im Sinne der Systemtheorie (Fourier-Transformation) beschreibbar sind.
Wegen des erhöhten Rechenaufwandes ist jeweils zu prüfen, ob nicht Simulation durch andere Verfahren sinnvoller ist. Trotz der hohen Rechengeschwindigkeit des ZENITH PC benötigen Rangordnungsfilter einige Minuten Rechenzeit. Im einzelnen ergeben sich folgende Zeiten:

Filtergröße	Sekunden
3 · 3	210
5 · 5	322
7 · 7	432

Im Gegensatz zur globalen Histogramm-Modifikation wird bei der lokalen Methode aus der Grauwerte-Statistik der Umgebung lediglich deren Mittelpunkt modifiziert. Hierbei wird das Histogramm auf eine gewünschte Form für die gesamte Umgebung angepaßt und nur der Mittelpunkt über Lookup-Tabelle besetzt.

Brauchbare Ergebnisse lassen sich erst bei Umgebungsgrößen ab 11 · 11 erzielen. Wegen der hiermit verbundenen Rechenzeiten ist auf eine Realisierung in PCBILD derzeit noch verzichtet worden.

Im nächsten mc-Heft soll gezeigt werden, wie mit Hilfe von Fouriertransformationen Störungen im Bild weitgehend behoben werden können. Dazu werden zunächst die mathematischen Grundlagen der FFT (Fast Fourier Transform) und deren Programmierung behandelt. Zum Beispiel sind Verwacklungen im Fourierspektrum durch orthogonale Streifen erkennbar. Defokussierungen führen zu konzentrischen Kreisen um den Mittelpunkt. Es werden Listings aus einem Programm-Paket in FORTRAN vorgestellt, die in erster Linie zum Verständnis der Algorithmen der Bildverarbeitung beitragen sollen und deshalb einfach und übersichtlich gehalten sind.

Bild 18/19/20. Kantendetektion im Original (links oben) durch den Sobel-Operator (rechts oben, Bild 18), eine Matrix zur Erkennung von senkrechten Kanten (links unten, Bild 19) und eine Matrix zur Erkennung von Kanten mit 45° Neigung (rechts unten, Bild 20)

Acer von Multitech

Optimieren auch Sie Ihren Gewinn mit hochwertigen, leistungsfähigen Computern und Peripherien, entwickelt und gefertigt von Multitech – für alle Anwendungen.

Der **Acer 500** („VOLKS-PC") hat sich vieltausendfach bewährt in den unterschiedlichsten Einsatzmöglichkeiten. Unser schnellster PC auf der Basis des 8088 ist der **Acer 710** mit einer Taktgeschwindigkeit von **10** MHz. Für preisbewußte Anwender der 80286-Technik bietet sich der **Acer 910** an. Weit in die Zukunft reichen die Möglichkeiten des **Acer 900** mit seiner großen Ausbaufähigkeit im AT-Bereich. Den vorläufigen Höhepunkt der PC Erfolgsmodelle stellt der 32-bit **Acer 1100** mit dem Hochleistungs-Mikroprozessor Intel **80386** und einer Taktgeschwindigkeit von 16 bis 4,77 MHz dar.

Neben diesen gewinnbringenden Computern vertreibt **CE-TEC** das gesamte Spektrum an Zubehör und möglichen Erweiterungen von verschiedenen Herstellern, um optimale Nutzung zu gewährleisten. Grafik-, Funktions- und Spezialkarten, Tastaturen und Massenspeicher gehören zum Lieferprogramm ebenso wie **Monitore,** Netzwerke **(LAN)** und der LASER-Drucker **Acer LP-75** u.v.a.m. Das komplette Programm aus einer Hand! Etwa 300 CE-TEC-Fachhändler in der Bundesrepublik und West-Berlin geben gern weitere Auskünfte.

Acer 1100

Acer 1100 B:
DM 8.995,–
unverbindliche Preisempfehlung

Acer 1100 B: CPU 80386, 16 MHz (schaltbar bis 4,77 MHz), 8 Steckplätze: 1 x für 32-bit, 5 x für 16-bit, 2 x für 8-bit. 1 MB RAM bis 16 MB. 1 FDD mit 1,2 MB (optional 40 MB-HD, 80 MB-HD [28 ms]), MS-DOS 3.2, Grafikkarte, 14" Monitor und Tastatur.

CE-TEC

Generalimporteur:
CE-TEC Trading GmbH · Kornkamp 4 · D-2070 Ahrensburg
Telefon: 04102/49 01-0 · Telex: 2 189 875 · Telefax: 04102/49 01 38

Hagen Völzke

Digitale Bilderzeugung und -verarbeitung

Teil 5: Die Software zur Grafikkarte

Im letzten Teil der Serie geht es um die Software, mit der man die Grafikkarte für den mc-68000-Computer aus dem vorangegangenen Heft und vor allem den darauf eingesetzten Grafikprozessor GDP 82786 anspricht – ein Beispiel, das sicher auch anderen GDP-Anwendern weiterhilft.

Vor einem näheren Eingehen auf die Software sind noch ein paar Worte zum Hardwareteil nötig. Zum Verdrahten des GDP 82786 ist große Aufmerksamkeit notwendig – Fehler beim Abzählen der Anschlußbeinchen sind schnell passiert. Die benötigten Bauteile sind in der *Tabelle* aufgeführt. Den Videotakt sollte auf jeden Fall ein Oszillator mit der doppelten Frequenz erzeugen, ein Flipflop teilt ihn dann auf die Nennfrequenz. Dadurch erhält man ein Taktsignal mit 50% Puls/Pause-Verhältnis.

Externe Synchronisation

Der GDP 82786 kann wahlweise auch extern synchronisiert werden. Es ist damit möglich, ein Computerbild mit einem externen Videosignal zu überlagern oder zu mischen. Das externe Videosignal kann z. B. von einer Kamera oder einem Videorecorder geliefert werden. Die horizontale Anlaufgenauigkeit des GDP 82786 liegt bei ±1 Pixel. Bei externer Synchronisation werden die Anschlüsse HSYNC und VSYNC zu Eingängen für die Synchronsignale des externen Videosignals.

Durch die externe Synchronisation können auch mehrere GDPs parallelgeschaltet werden, z. B. um höhere Bittiefen zu realisieren. Werden alle GDPs synchron mit dem selben Takt gespeist, so treten keine Abweichungen beim Anlaufen auf.

In einer weiteren Betriebsart kann der Prozessor auf den Leitungen HSYNC und VSYNC während der aktiven Displayphase Informationen ausgeben, ob er sich gerade in einem Fenster befindet. So lassen sich z. B. in verschiedenen Fenstern verschiedene Farbpaletten benutzen und damit die Zahl der möglichen Farben steigern. Konkrete Realisierungshinweise zu diesen Betriebsarten mit Applikationen geben die Datenblätter des GDP 82786. Eine wertvolle Hilfe zur Entwicklung eigener Schaltungen stellt der Applikationsbericht AP 270 „82786 Hardware Configuration" dar.

Software

Die Programmierung des GDP 82786 ist im Prinzip recht einfach, die Tücke steckt im Detail. Aus diesem Grund finden Sie in *Bild 1* ein komplettes Treiberprogramm unter CP/M-68k. Anhand dieses Beispiels kann man am besten verstehen, wie die einzelnen Parameter und Register des GDP eingestellt werden. Das Programm beinhaltet die komplette Initialisierung des GDP 82786 sowie einen speicherresidenten Treiber, der von Assembler und C angesprochen werden kann. Das Programm verschiebt sich dazu nach dem Start ans Ende der TPA und schützt sich dann vor Überschreiben durch Herabsetzen der oberen TPA-Grenze. Über

```
*************************************************************************
*                                                                       *
*      Einfacher Treiber für den Intel-Graphiprozessor GDP 82786        *
*               Erstellt am 6.12.1987 von H. Völzke                     *
*                                                                       *
*************************************************************************

* System Definitionen

slottab     equ     $fff800                 ; Basisadresse der SIottabelle
kennung     equ     $07028000               ; Kennung der Graphikkarte
intvec      equ     $80                     ; Interruptvektor

***********************
* variable Definitionen

dvideo      equ     0                       ; Default Video Farbe

*********************************************
* Speicher Definitionen für 256 KByte Video-RAM
* Der Graphiksppeicher von 256 KByte wird wie
* folgt aufgeteilt:
* $00000 ... $03FFF:  16 KByte Puffer für Displayprozessor
* $04000 ... $07FFF:  16 KByte Puffer für Graphikprozessor
* $08000 ... $3FFFF: 224 KByte Bildspeicher

ldall       equ     $00000                  ; Speicher für GDP Parameter
ldlow       equ     $0000
ldup        equ     $0                      ; aufgeteilt in Worte

combuf      equ     $04000                  ; Speicher für Plotkommandos
comlow      equ     $4000
comup       equ     $0

framebuf    equ     $08000                  ; Bildspeicher
framelow    equ     $8000
frameup     equ     $0

gdpoff      equ     $100000                 ; Offset für GDP I/O-Bereich
clutoff     equ     $100100                 ; Offset für CLUT

*********************************************
* GDP Definitionen, Offsets für interne Register

intrel      equ     0
biuctr      equ     4
refrctr     equ     6
```

Bild 1: Das Grundgerüst eines Treibers unter CP/M-68k – leicht zu erweitern und als Beispiel sicher aussagekräftiger als viele Worte

```
dramctr     equ     8
disprio     equ     10
graprio     equ     12
extprio     equ     14
gpopcode    equ     $20
gplinklo    equ     $22
gplinkup    equ     $24
gpstat      equ     $26
gpinsplo    equ     $28
gpinspup    equ     $2A
dpopcode    equ     $40
dppara1     equ     $42
dppara2     equ     $44
dppara3     equ     $46
dpstat      equ     $48
defvdata    equ     $4A
***************
* GDP Kommandos

C_LINK      equ     $200                ; LINK-Befehl
C_LDALL     equ     $500                ; LOADALL-Befehl
****************
* Programm Start

            lea     slottab,a0          ; Karte in Slottabelle suchen
            move.w  #23*16,d7
look        cmpi.l  #kennung,0(a0,d7.w)
            beq.s   found
            sub.w   #16,d7
            bpl.s   look

            lea     nfnderr(pc),a0      ; falls nicht gefunden, dann
            move.l  a0,d1               ; Fehlermeldung ausgeben
            moveq   #9,d0
            trap    #2

            clr.w   d0                  ; und abbrechen
            trap    #2

* Das Programm verschiebt nun den später residenten Teil an das
* Ende der TPA und setzt die TPA-Grenzen herunter. Interrupt-
* routine und Graphikfunktionen bleiben somit ständig verfügbar.
* In der Slottabelle wird die Adresse der Einsprungtabelle für
* die Graphikfunktionen abgespeichert.

found       moveq   #$3f,d0             ; get TPA limits
            lea     tpab(pc),a2
            move.l  a2,d1
            trap    #2

            lea     start(pc),a4        ; Start address to relocate
            movea.l 6(a2),a3            ; TPA high limit
            suba.l  a2,a3               ; subtract program length
            adda.l  a4,a3               ; tpab address is end
            move.l  a3,6(a2)            ; restore new limit

            move.w  #%11,(a2)           ; function 'set tpa limits'
            moveq   #$3f,d0
            trap    #2

            sub.l   a4,d1               ; get program length

            movea.l a3,a2               ; A3 is new start address
copy        move.b  (a4)+,(a2)+         ; transfer text to new
            subq.l  #1,d1               ; address
            bne.s   copy
            jmp     (a3)                ; now jump to new address

* Der letzt JMP-Befehl läuft bei 'start' weiter, allerdings
* bereits unter der neuen Adresse des Programmes am Ende der TPA.
```

```
start       move.w  #intvec,$fffff4     ; setup IRQ vector-#
            lea     irq(pc),a1
            move.l  a1,intvec*4
            lea     variables(pc),a6    ; get var-pointer

            lea     entry(pc),a1        ; Wichtig: Alles PC-relativ!
            move.l  a1,8(a0,d7.w)       ; setup global pointer
            lea     ginit(pc),a2
            move.l  a2,(a1)+
            lea     move_abs(pc),a2
            move.l  a2,(a1)+
            lea     move_rel(pc),a2
            move.l  a2,(a1)+
            lea     line(pc),a2
            move.l  a2,(a1)+
            lea     circle(pc),a2
            move.l  a2,(a1)+
            lea     rectan(pc),a2
            move.l  a2,(a1)+

            movea.l 4(a0,d7.w),a0       ; get GDP base
            movea.l a0,a1
            adda.l  #gdpoff,a1          ; A1 ist Basis für I/O-Bereich
            move.l  a0,base(a6)         ; setup base for irq
            clr.w   busy(a6)

            move.b  #$30,biuctr+1(a1)   ; Biu Control (lower Byte!)
            move.b  #0,biuctr(a1)       ; Zugriff Byteweise!
            move.w  #0,(a1)             ; relocation := 0, I/O-mapped
            move.w  #%1011,dramctr(a1)  ; setup DRAM (page mode)
            move.w  #18,refrctr(a1)     ; Refreshzyklus 4 ms
            move.w  #%111110,disprio(a1) ; Prioritäten für Zugriffe
            move.w  #%100011,graprio(a1) ; von CPU, Plotprozessor und
            move.w  #%101000,extprio(a1) ; Displayprozessor

            lea     allregs(pc),a2      ; jetzt alle Register laden
            movea.l a0,a3               ; dazu zunächst
            adda.l  #ldall,a3           ; alles in den Speicher des
            move.w  #datalen/2-1,d0     ; GDP kopieren
loop        move.w  (a2)+,(a3)+
            dbra    d0,loop

            move.w  #ldlow,dppara1(a1)
            move.w  #ldup,dppara2(a1)   ; Start address at <ldall>
            move.w  #C_LDALL,dpopcode(a1) ; LOAD all

            move.w  #dvideo,defvdata(a1) ; default video data

            movea.l a0,a2               ; jetzt die CLUT initialisieren
            adda.l  #clutoff,a2         ; bump to clut
            move.b  #$ff,5(a2)          ; set pixel mask
            moveq   #0,d0
            move.b  #0,1(a2)            ; ab Adresse 0

lop         move.b  d0,d1               ; Graustufen von 0 ... 63
            and.b   #$3f,d1             ; viermal wiederholen auf den
            move.b  d1,3(a2)            ; logischen Farben 0 ... 255
            move.b  d1,3(a2)
            move.b  d1,3(a2)
            addq.b  #1,d0
            bne     lop

            movea.l a0,a1               ; now clear memory
            adda.l  #framebuf,a1
            move.w  #$38000/4-1,d0      ; den angezeigten Bildspeicher
            move.l  #$01010101,d1       ; mit Farbe '01' füllen
cl0         move.l  d1,(a1)+
            dbra    d0,cl0

            clr.w   d0                  ; zurück ins CP/M, Treiber
            trap    #2                  ; ist fertig initialisiert
```

```
*******************
* Interrupt Service

irq         pea     (a0)                    ; Register retten
            pea     (a1)
            lea     variables(pc),a1        ; Pointer auf Variablen
            movea.l base(a1),a0             ; get gdp base
            adda.l  #gdpoff,a0              ; bump to gdp i/o
            tst.w   gpstat(a0)              ; reset interrupt flags
            tst.w   biuctr(a0)              ; clear interrupt pending
            clr.w   busy(a1)                ; service system flag
            movea.l (a7)+,a1                ; restore registers
            movea.l (a7)+,a0
            rte

****************
* Plotfunktionen
* Es folgen nun einige Plotfunktionen. Sie können von Assembler
* oder C aus angesprochen werden. Dazu werden zunächst alle
* Befehle im Graphikspeicher gefolgt von einem HALT-Befehl
* angelegt. Beim Aufruf des jeweiligen Befehls werden dann
* lediglich die Parameter im GDP-Speicher eingetragen und
* der Graphikprozessor wird gestartet.
* Es werden die Adressregister A0 und A1 sowie das Datenregister
* D0 benutzt. Diese sind unter C Scratchregister und müssen
* daher nicht gerettet werden.

ginit       lea     variables(pc),a0        ; Initialisierung
gin1        tst.w   busy(a0)                ; wait for GDP ready
            bne.s   gin1
            st      busy(a0)                ; set flag: GDP running
            movea.l base(a0),a1             ; command buffer in GDP memory
            adda.l  #combuf,a1
            lea     gintab(pc),a0           ; initialize command table
            move.w  #ginlen/2-1,d0
gin0        move.w  (a0)+,(a1)+
            dbra    d0,gin0
            lea     variables(pc),a0
            movea.l base(a0),a0
            adda.l  #gdpoff,a0
            move.w  #comlow,gplinklo(a0)
            move.w  #comup,gplinkup(a0)
            move.w  #C_LINK,gpopcode(a0)    ; Starte init-Befehls-Sequenz
            rts                             ; (siehe 'gintab')

move_abs    lea     variables(pc),a0
            movea.l base(a0),a1
            adda.l  #combuf,a1
            adda.l  #amovop-gintab,a1
ma1         tst.w   busy(a0)
            bne.s   ma1
            st      busy(a0)
            movea.l base(a0),a0
            adda.l  #gdpoff,a0
            move.w  4(a7),2(a1)             ; dx und dy vom Stack in den
            move.w  6(a7),4(a1)             ; GDP-Speicher umkopieren
            move.w  #comlow+amovop-gintab,gplinklo(a0)
            move.w  #comup,gplinkup(a0)
            move.w  #C_LINK,gpopcode(a0)
            rts
move_rel    lea     variables(pc),a0
            movea.l base(a0),a1
            adda.l  #combuf,a1
            adda.l  #rmovop-gintab,a1
mr1         tst.w   busy(a0)
            bne.s   mr1
            st      busy(a0)
            movea.l base(a0),a0
            adda.l  #gdpoff,a0
            move.w  4(a7),2(a1)
            move.w  6(a7),4(a1)
            move.w  #comlow+rmovop-gintab,gplinklo(a0)
            move.w  #comup,gplinkup(a0)
            move.w  #C_LINK,gpopcode(a0)
            rts

line        lea     variables(pc),a0
            movea.l base(a0),a1
            adda.l  #combuf,a1
            adda.l  #lineop-gintab,a1
l1          tst.w   busy(a0)
            bne.s   l1
            st      busy(a0)
            movea.l base(a0),a0
            adda.l  #gdpoff,a0
            move.w  4(a7),2(a1)
            move.w  6(a7),4(a1)
            move.w  #comlow+lineop-gintab,gplinklo(a0)
            move.w  #comup,gplinkup(a0)
            move.w  #C_LINK,gpopcode(a0)
            rts

circle      lea     variables(pc),a0
            movea.l base(a0),a1
            adda.l  #combuf,a1
            adda.l  #circop-gintab,a1
c1          tst.w   busy(a0)
            bne.s   c1
            st      busy(a0)
            movea.l base(a0),a0
            adda.l  #gdpoff,a0
            move.w  4(a7),2(a1)
            move.w  #comlow+circop-gintab,gplinklo(a0)
            move.w  #comup,gplinkup(a0)
            move.w  #C_LINK,gpopcode(a0)
            rts

rectan      lea     variables(pc),a0
            movea.l base(a0),a1
            adda.l  #combuf,a1
            adda.l  #rectop-gintab,a1
rec1        tst.w   busy(a0)
            bne.s   rec1
            st      busy(a0)
            movea.l base(a0),a0
            adda.l  #gdpoff,a0
            move.w  4(a7),2(a1)
            move.w  6(a7),4(a1)
            move.w  #comlow+rectop-gintab,gplinklo(a0)
            move.w  #comup,gplinkup(a0)
            move.w  #C_LINK,gpopcode(a0)
            rts
***********
* Variablen

variables   ds.l    1                       ; Speicherplatz für Variablen
            ds.w    1
entry       ds.l    6                       ; 6 Funktionseinsprünge

base        equ     0                       ; Offsets für Variablen
busy        equ     4

*******************
* Daten Definitionen
* Es folgen Video-Parameter für folgende Konfiguration:
* Pixeltakt: 16 MHz, 2 interleaved RAM-Bank's
* Auflösung: 640 x 250 Punkte, 8 Bit pro Punkt
* Ablenkfrequenzen: horizontal 15.625 Hz, vertikal: 50 Hz
```

```
HFldStrt    equ     240                     ; Beginn der Zeile
VFldStrt    equ     35                      ; Beginn des Bildes

allregs     dc.w    3                       ; status
            dc.w    $ff                     ; IntMask
            dc.w    16                      ; trip point
            dc.w    0                       ; frame interrupt
            dc.w    0                       ; reserved
            dc.w    0                       ; crtmode (non interlaced)
            dc.w    100                     ; HSynStp
            dc.w    HFldStrt                ; HFldStrt
            dc.w    HFldStrt+640            ; HFldStp (16 MHz, 640 Dots)
            dc.w    1024                    ; Linelen (15.625 Hz)
            dc.w    10                      ; VSynStp
            dc.w    VFldStrt                ; VFldStrt
            dc.w    VFldStrt+250            ; VFldStp
            dc.w    312                     ; FramLen (50 Hz)
            dc.w    ldlow+window-allregs    ; Offset do window descr.
            dc.w    ldup                    ; upper word
            dc.w    0                       ; reserved
            dc.b    0                       ; X-Zoom
            dc.b    0                       ; Y-Zoom
            dc.w    0                       ; Fieldcolor
            dc.w    $FF                     ; Bordercolor (Fenster-Rahmen)
            dc.w    $FE                     ; obere Pixel für 1 Bit/Punkt
            dc.w    $FC                     ; 2 Bit/Punkt
            dc.w    $F0                     ; 4 Bit/Punkt
            dc.b    $a0                     ; Cursor Mode 16x16, transparen

            dc.b    $fe                     ; höhere Bits der Cursor Farbe
            dc.w    HFldStrt+10             ; Cursor X
            dc.w    VFldStrt+10             ; Cursor Y
            dc.w    %0000000000000000       ; Cursor Pattern
            dc.w    %1111111111111110       ; 'Sanduhr'
            dc.w    %0100000000000100
            dc.w    %0010000000001000
            dc.w    %0001000000010000
            dc.w    %0000101110100000
            dc.w    %0000011111000000
            dc.w    %0000001110000000
            dc.w    %0000000100000000
            dc.w    %0000010001000000
            dc.w    %0000100000100000
            dc.w    %0001001110010000
            dc.w    %0010111111101000
            dc.w    %0111111111111100
            dc.w    %1111111111111110
            dc.w    %0000000000000000

  * Es folgt ein Windowdeskriptor mit 1 Strip und 1 Tile

  window    dc.w    249                     ; 250 Zeilen (n-1)
            dc.w    ldlow+window-allregs    ; Rückwärts-Link auf
            dc.w    ldup                    ; sich selbst
            dc.w    0                       ; 1 Tile (n-1)
            dc.w    640                     ; Bit-Map-Width
            dc.w    framelow                ; Startadresse des
            dc.w    frameup                 ; Bildspeichers
            dc.w    $8F0                    ; Bpp/Start/Stopbit
            dc.w    638                     ; Fetch-Count
            dc.w    $F000                   ; Rahmen an allen 4 Kanten

  ende      ds.b    0
  datalen   equ     ende-allregs

  **********************
  * Zeichen-Roh-Befehle

gintab      dc.w    $1A00                   ; Define_Bit_Map
            dc.w    framelow                ; Startadresse bei framebuf
            dc.w    frameup
            dc.w    639                     ; maximale X-Koordinate
            dc.w    249                     ; maximale Y-Koordinate
            dc.w    8                       ; 8 Bit/Pixel

            dc.w    $4100                   ; Define_Logical_Op
            dc.w    $FFFF                   ; Farbmaske
            dc.w    %110                    ; XOR-Betriebsart

            dc.w    $3D00                   ; Define_Colors
            dc.w    $FFFF                   ; Vordergrundfarbe
            dc.w    0                       ; Hintergrundfarbe

            dc.w    $600                    ; Define_Texture
            dc.w    $FFFF                   ; Muster

            dc.w    $4600                   ; Define_Clipping_Rectangle
            dc.w    0                       ; Xmin = 0
            dc.w    0                       ; Ymin = 0
            dc.w    639                     ; Xmax = 639
            dc.w    249                     ; Ymax = 249

            dc.w    $4F00                   ; Move_Abs
            dc.w    319                     ; auf Bildschirm-Mitte
            dc.w    124                     ; positionieren

            dc.w    $3400                   ; Load IM
            dc.w    comlow+imadr-gintab     ; Interruptregister
            dc.w    comup                   ; laden
            dc.w    4

            dc.w    1                       ; HALT

amovop      dc.w    $4F00                   ; Rohbefehl für Move_abs
            dc.w    0
            dc.w    0
            dc.w    1                       ; HALT-Befehl

rmovop      dc.w    $5200                   ; Move_rel
            dc.w    0
            dc.w    0
            dc.w    1
lineop      dc.w    $5400                   ; Line
            dc.w    0
            dc.w    0
            dc.w    1

circop      dc.w    $8E00                   ; Circle
            dc.w    0
            dc.w    1

rectop      dc.w    $5800                   ; Rectangle
            dc.w    0
            dc.w    0
            dc.w    1

imadr       dc.w    $7F                     ; Interrupt mask (POLL)
ginlen      equ     *-gintab

  * bis hierher bleibt alles resident in der TPA
  * ------------------------------------------------------------------------

tpab        dc.w    0                       ; Puffer für TPA-Grenzen
            ds.l    2
nfnderr     dc.b    "GDP-Karte nicht in Slottabelle eingetragen!",13,10,'$'


  ************************************ EOF ************************************
*
```

eine Sprungtabelle, deren Anfangsadresse in der Slottabelle vermerkt wird, können andere Programme auf Funktionen des Treibers zugreifen. Zur Initialisierung der Grafikkarte sind folgende Schritte nötig:

1. Montage der Grafikkarte in einem freien Steckplatz des mc-68000-Computers, wobei nicht die Steckplätze 0, 1 oder nach einer RAM-Karte verwendet werden dürfen
2. Einschalten des Computers
3. Anpassen der Datei GDPSLOT.S *(Bild 2)* an den gewählten Steckplatz
4. GDPSLOT.S neu übersetzen
5. Aufruf des Programmes GDPSLOT.68K
6. Treiber GDP.S übersetzen und GDP.68K starten

Jetzt sollte sich auf dem Bildschirm ein Fenster mit weißem Rahmen, leicht grauem Untergrund und einer Sanduhr in der linken oberen Ecke zeigen. Das Programm GDP.S ist für einen Standardbildschirm und eine Punktfrequenz von 16 MHz ausgelegt. GDP.68K sollte nur ein einziges Mal aufgerufen werden. Falls der Grafikprozessor einmal abstürzt, ist ein Neustart des Computers durch Reset sicherer.

Das Arbeitsprinzip des Treibers ist sehr einfach: Soll ein bestimmter Grafikbefehl ausgeführt werden, wird er zusammen mit den jeweiligen Parametern und gefolgt von einem HALT-Befehl in den Arbeitsspeicher des GDP 82786 übertragen. Dann erhält der Plotprozessor das Kommando, den Befehl im Speicher auszuführen. Um eine effiziente Kommunikation zwischen GDP und CPU 68000 zu ermöglichen, löst der GDP nach Beendigung der Befehlsabarbeitung einen Interrupt aus. Der mc-68000-Computer weiß dann, das der GDP mit dem letzten Befehl fertig ist und merkt sich dies in einem Flag im Speicher. Prinzipiell kann man auch das Statusregister des GDP abfragen, doch funktioniert das nicht so zuverlässig wie die Interrupt-Methode. Das Prin-

```
*************************************
* GDPSLOT.S: Eintragen der Graphikkarte
*            in die Slottabelle und
*            Programmierung des
*            Adressdekoders

slot        equ     2
slottab     equ     $fff800
kennung     equ     $07028000

            move.l  #kennung,slottab+slot*16  * Kennung eintragen
            moveq   #$3e,d0                   * Set to Supervisor
            trap    #2
            jsr     $ff745c                   * Unterprogramm
            clr.w   d0                        * Adressdekoder setzen
            trap    #2                        * Warmstart ins CP/M
```

Bild 2: Hier wird's „mc-68000-Computer-spezifisch": Das Programm GDPSLOT.S erledigt die Eintragung in die Slottabelle

```
/*********************************************
*                                            *
*      Header-Datei fuer den GDP 82786       *
*                                            *
*********************************************/

extern ginit(),move_abs(),move_rel(),line(),circle(),rectangle();

int     *c_slot,*getslot();          /* globale Variable, sichert Adresse
                                        der Einsprungtabelle */

gdpinit()
{
        if ((c_slot = getslot()) == 0)
        {
            printf("Treiber nicht installiert!\n");
            return(-1);
        }
        else
        {
            ginit();
            return(0);
        }
}

/* gdpinit setzt die Variable c_slot, ueber die der Treiber angesprungen
   werden kann. Ist die Karte nicht in der Slottabelle eingetragen oder
   der Treiber nicht installiert worden, so gibt gdpinit eine entsprechende
   Fehlermeldung aus und gibt -1 als Fehlercode zurueck.
   Bei Erfolg wird die 'ginit'-Funktion aufgerufen und 0 zurueckgegeben */
```

Bild 3: Mit Header-Dateien kann man unter C viel Arbeit sparen – hier die Datei GDP.H, die das Treiberprogramm unter CP/M-68K sucht

Tabelle 1: Stücklisten

Grafikkarte			
Stück	Typ	Position	Alternativ
1	GDP 82786 (Intel)	IC1	
8	DRAM 41464-12	IC2...IC9	siehe Text
1	7407	IC19	
1	74 LS 08	IC15	
1	74 ALS 04	IC18	74 AS/S 04
2	74 LS 32	IC16, IC17	
1	74 AS 74	IC14	74 S 74
1	74 LS 122	IC13	
1	74 LS 138	IC12	
2	74 ALS 646	IC10, IC11	74 xx 646
1	Quarzoszill.	X1	20 MHz

Lookuptable mit IMS G170			
Stück	Typ	Position	Alternativ
1	IMS G170-35	IC20	
1	LM 334	IC21	
1	74 AS 08	IC23	74 ALS/S 08
1	74 ALS 86	IC22	74 S 86
1	74 AS 244	IC24	74 S 244
1	Quarzoszillator	X2	Pixeltakt x 2
1	Spule 100 µH	L1	
1	Widerstand	33 Ω Me	
1	Widerstand	330 Ω Me	
diverse Kondensatoren			

```
*****************************************
* GDPINCL.S: Externe Assemblerprozeduren
* zum Ansteuern der Graphikkarte mit
* dem GDP 82786. Die Variable _c_slot
* muß vor Aufruf der Graphikfunktionen
* durch die Funktion 'gdpinit()' definiert
* worden sein (siehe 'gdp.h')

.globl _getslot,_ginit,_move_abs,_move_rel,_line,_circle,_rectangle
.globl _c_slot

_getslot    lea     $fff800,a0              ; Karte in Slottabelle suchen
            move.w  #23*16,d0
look        cmpi.l  #$07028000,0(a0,d0.w)
            beq.s   found
            sub.w   #16,d0
            bpl.s   look
bad         moveq   #0,d0                   ; Fehler: Nil zurueckgeben
            rts
found       move.l  8(a0,d0.w),d0           ; Adresse zurueckgeben
            cmpi.l  #$ffffffff,d0           ; ungueltig?
            beq.s   bad
            rts

_ginit      movea.l _c_slot,a0
            movea.l (a0),a0
            jmp     (a0)

_move_abs   movea.l _c_slot,a0
            movea.l 4(a0),a0
            jmp     (a0)

_move_rel   movea.l _c_slot,a0
            movea.l 8(a0),a0
            jmp     (a0)

_line       movea.l _c_slot,a0
            movea.l $C(a0),a0
            jmp     (a0)

_circle     movea.l _c_slot,a0
            movea.l $10(a0),a0
            jmp     (a0)

_rectangle  movea.l _c_slot,a0
            movea.l $14(a0),a0
            jmp     (a0)
```

Bild 4: Diese externen Assembler-Prozeduren koppeln C-Programme an den Grafik-Treiber

```
/*****************************************
*                                        *
*        Einfaches Testprogramm          *
*                                        *
* Es werden einige Kreise und ineinander *
* geschachtelte Rechtecke gezeichnet     *
*****************************************/

#include <gdp.h>


main()
{
int i;

  if ( gdpinit() == 0 )                 /* Nur falls Initialisierung */
  {                                     /* erfolgreich war !         */
    move_abs(125,125);

    for (i = 0; i <= 120; i += 5 )      /* konzentrische Kreise zeichnen */
        circle(i);

    for (i = 0; i <= 200; i += 10 )     /* Geschachtelte Rechtecke */
    {
       move_abs(450-i/2,125-i/2);
       rectangle(i,i);
    }
  }
}
```

Bild 5: Ein einfaches Testprogramm: TEST.C

zip nutzt natürlich nicht die volle Leistung des GDP, da dieser immer nur einen Befehl ausführt und dann wieder stehen bleibt. Im vorliegenden Treiber sind außerdem nur fünf Grafikbefehle implementiert: (Move_Abs,Move_Rel, Line, Circle und Rectangle): Es soll hier ja nur ein Grundgerüst gezeigt werden. Immerhin kann man mit diesen 5 Befehlen schon arbeiten und die meisten anderen lassen sich einfach implementieren.

Ansteuerung von C aus

Ein Ansprechen der Grafik von der Sprache C aus ist nicht schwierig: Eine Header-Datei *(Bild 3)* deklariert die externen Prozeduren. Außerdem wird eine globale Variable „c_slot" definiert, in der die Anfangsadresse der Einsprungtabelle gespeichert wird. Die Funktion gdpinit() prüft zunächst, ob die Grafikkarte und der Treiber GDP.68K ordnungsgemäß installiert sind und initialisiert dann den Plotprozessor (Funktion ginit). Ist die Karte nicht installiert oder fehlt der Treiber GDP.68K, so liefert getslot einen Fehler zurück. Die Funktion gdpinit gibt diesen Fehler an das aufrufende Programm weiter. *Bild 4* zeigt die externen Assemblerprozeduren, die direkt auf den im Speicher residenten Treiber zugreifen. Die Datei GDPINCL.S muß übersetzt und beim Linken als GDPINCL.O mit eingebunden werden. *Bild 5* zeigt ein kleines Testprogramm, das in der linken Bildhälfte konzentrische Kreise zeichnet und in der rechten Bildhälfte ineinandergeschachtelte Rechtecke. Man sieht: Auf Hochsprachen-Ebene lassen sich die Grafikfunktionen sehr komfortabel anwenden. In *Bild 6* schließlich sind die nötigen Übersetzungsschritte für die Assembler- und C-Programme aufgeführt.

Die Register des GDP 82786

Bild 7 zeigt den Register-Block des Displayprozessors im GDP 82786. Die meisten Parameter sind eigentlich selbsterklärend, daher folgt nur eine kurze Schilderung einiger Besonderheiten:

Der „Trip-Point" bestimmt das Zugriffsverhalten des Pixel-FIFOs auf den Bildspeicher. Der Wert legt fest, wieviel Worte minimal im FIFO sein dürfen, bevor ein neuer Ladezugriff stattfindet. Ein niedriger Wert nutzt die Geschwindigkeit der RAM-Bausteine besser aus, da immer große Blöcke auf einmal geladen werden, dafür ist die Warte-Wahrscheinlichkeit für Zugriffe der Haupt-CPU größer.

Der Parameter Frint läßt Interrupts erst nach <Frint> Bilddurchläufen zu. Damit kann z. B. Blinken per Software realisiert

oder zwischen verschiedenen Bildern umgeschaltet werden.
Die Felder „1 Bpp Pad", „2 Bpp Pad" und „4 Bpp Pad" legen jeweils die oberen Bits der Videoausgänge des GDP 82786 fest, wenn Fenster mit weniger als 8 Bit pro Punkt angezeigt werden. Diese Parameter sind global für das gesamte Bild.

Beim „CsrPad" legen die oberen 7 Bit die Farbe des Cursors fest, das unterste Bit wird durch das Cursor-Muster selbst festgelegt. Dabei kann der Cursor wahlweise transparent sein (0-Bits lassen den Untergrund durchscheinen) oder „opaque", d.h. man kann durch Lücken im Cursor nicht hindurch schauen.

In *Bild 8* sieht man, wie die einzelnen Parameter im Registerblock des Displayprozessors auf das Bildformat Einfluß nehmen. Wichtig ist hier eigentlich nur, daß die Zeilen in Einheiten von Punkt-Takten zählen und die Bilder in Einheiten von Zeilen. Im Beispielprogramm wird der Parameter HSynStp mit 100 spezifiziert, d. h. ein Synchronimpuls dauert 100 Punkt-Takte oder – bei 16 MHz – 6,25 µs. Das ist eigentlich ein bißchen zu lang (das BAS-Signal schreibt weniger vor), aber man muß die gesamten Parameter ohnehin noch an den verwendeten Bildschirm anpassen.

Bild 9 schließlich zeigt den Aufbau eines „Window-Strip-Descriptors". Der erste Parameter gibt die Zahl der Zeilen in diesem Strip an (Zeilenzahl – 1), der zweite ist ein Zeiger auf den nächsten Strip-Deskriptor. Der letzte Strip-Deskriptor muß wieder auf den ersten zeigen, damit der Display-Prozessor beim nächsten Bild wieder von vorne anfangen kann. Jetzt folgt die Zahl der Tiles innerhalb dieses Strips. Auch hier wird wieder n-1 angegeben. Der Parameter „Bit Map Width" legt fest, um wieviel Byte der Adreßzähler, der den Bildspeicher ausliest, von Zeile zu Zeile erhöht wird. „Mem Start" definiert die Anfangsadresse der Bit-Map für das jeweilge Tile. Im folgenden Wort werden die Bittiefe (Bpp) sowie die Start- und Stop-Bit-Position innerhalb der Bitmap festgelegt. Dadurch, daß die Start- und Stop-Positionen bitgenau definiert werden können, sind Softscrolling und Panning möglich. Der Fetch-Count sagt dem Prozessor, wieviele Bytes für eine Linie aus dem Speicher zu holen sind. Man beachte, daß dieser Wert kleiner als die Bit-Map-Width sein kann, nämlich dann, wenn man nur einen kleinen Ausschnitt aus einem großen Bild anzeigt. Das letzte Wort eines Tile definiert über 4 Bits, an welchen Kanten ein Rahmen erzeugt wird. Der Rahmen liegt stets innerhalb eines Tile und belegt

```
; Uebersetzung der C-Programme

CP68 -I 0: TEST.C TEST.I
C068 TEST.I TEST.1 TEST.2 TEST.3 -F
ERA  TEST.I
C168 TEST.1 TEST.2 TEST.S
ERA  TEST.1
ERA  TEST.2
AS68 -U -L TEST.S
LO68 -S -T500 -U__OPTOFF -O TEST.68K S.O TEST.O GDPINCL.O CLIB LIBF.A

; Uebersetzung der Assembler Programme

AS68 -L GDPSLOT.S
LO68 -S -T500 -O GDPSLOT.68K GDPSLOT.O

AS68 -L GDP.68K
LO68 -S -T500 -O GDP.68K GDP.O

AS68 -L -U GDPINCL.S
```

Bild 6: So werden die C- und Assembler-Programme übersetzt und zusammengebunden

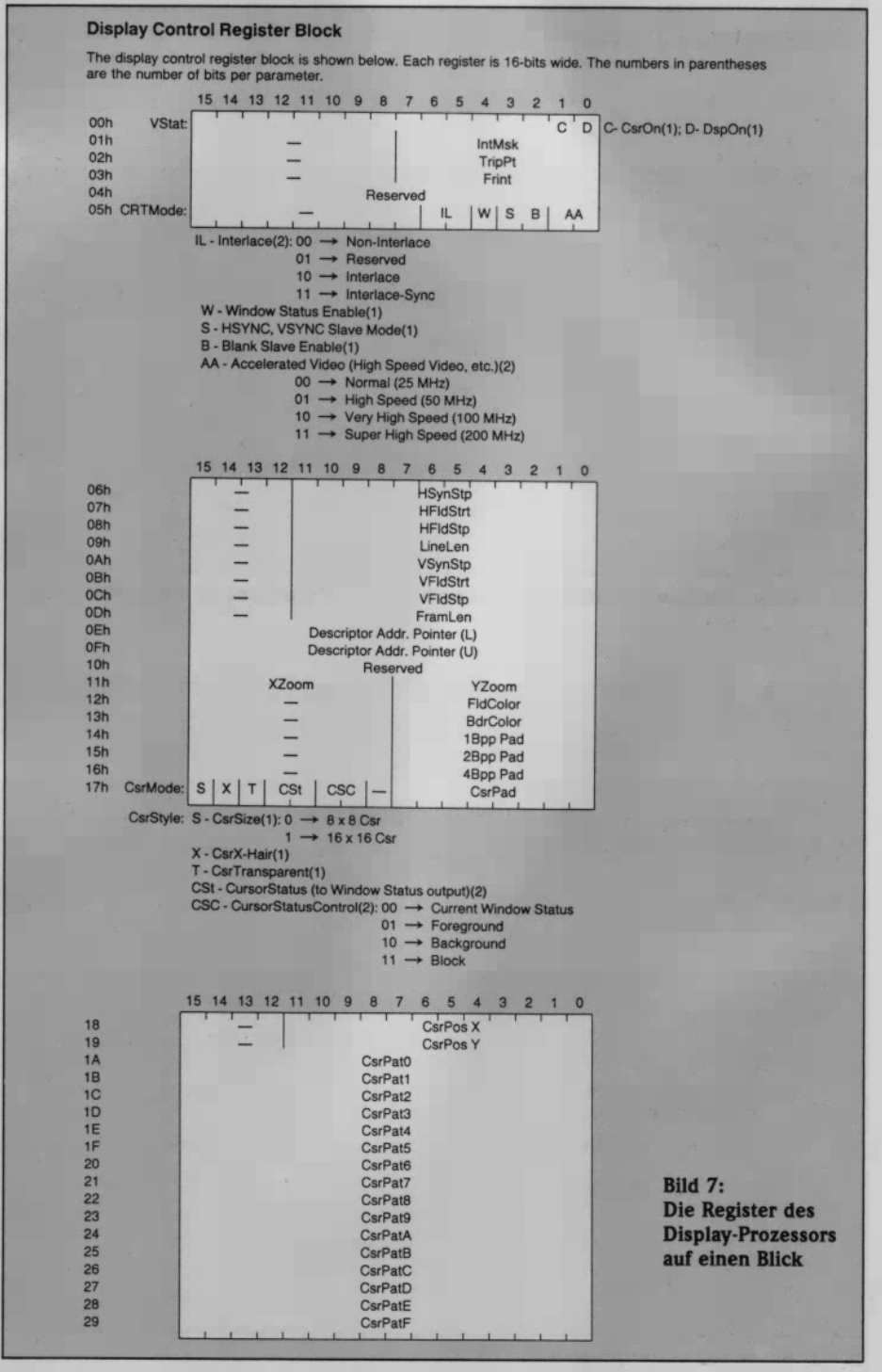


Bild 7: Die Register des Display-Prozessors auf einen Blick

ein Pixel. Die Farbe ist global für das ganze Bild durch die Border-Farbe einstellbar. Das Z-Bit sagt, ob das Tile vergrößert (Zooming) oder normal dargestellt werden soll. Die Zoomingfaktoren für X- und Y-Richtung lassen sich getrennt einstellen, gelten jedoch für alle Fenster gleichermaßen. Das F-Bit entscheidet, ob das Tile ein normaler Bildbereich ist, in dem Bildinformation dargestellt wird, oder ein „Feld"-Bereich, der lediglich eine Fläche zur Abgrenzung darstellt. Feldbereiche haben die Bittiefe 0, benötigen also keinen Grafikspeicher. Ihre Farbe kann global im Register-Block des Displayprozessors festgelegt werden. Ist das F-Bit gesetzt, so wird im Wort, in dem normalerweise Start- und Stop-Bit sowie Bpp angegeben werden, die Breite des Feldes in Punkten angegeben. Ein Feld besitzt keinen Speicher, daher sind die Parameter Fetch-Count und MemStart nicht sinnvoll und sollten auf 0 gesetzt werden. Bei der Programmierung ist zu beachten, daß der Fetchcount sowie erlaubte Start- und Stop-Bit-Positionen von der jeweiligen Bittiefe des Fensters abhängen.

Alle Adreßangaben beziehen sich stets auf den Grafikspeicher des GDP und zählen ab Adresse Null, unabhängig von der Lage im Adreßraum der CPU 68000. Im Beispiel mit 256 KByte Bildspeicher sind also Adressen im Bereich 0...$3FFFF sinnvoll.
Sicherlich sind viele Fragen über den Grafikprozessor mit diesem Beitrag nicht geklärt. Das Studium der Datenblätter ist unbedingt notwendig, dieser Artikel ist aber sicher eine wertvolle Ergänzung, da er viele Informationen enthält, die man in den Datenblättern mühsam „zwischen den Zeilen" lesen muß.

Die Serie „Digitale Bilderzeugung und -verarbeitung" ist damit abgeschlossen.

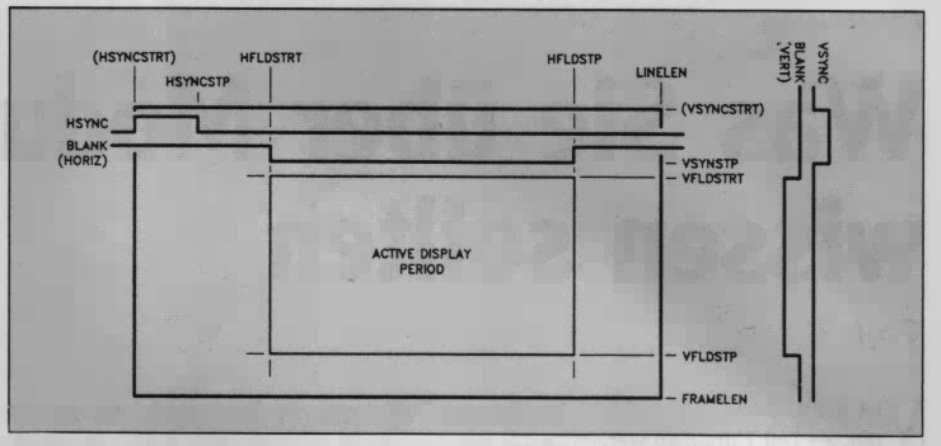


Bild 8: Die Abhängigkeiten der Bildsignale

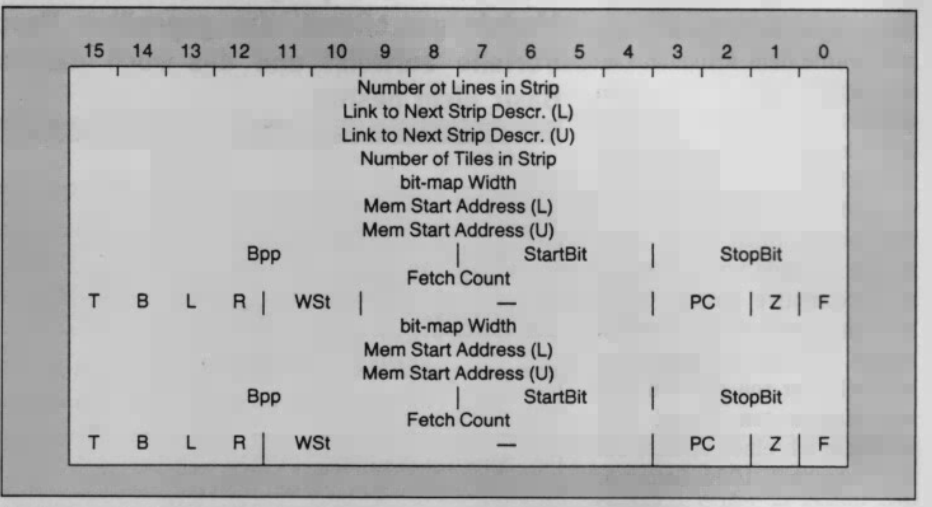


Bild 9: Der Aufbau der Window-Deskriptoren – die Bedeutung einzelner Werte ist im Text beschrieben

Lexikon der verwendeten Begriffe:

Begriff	Erklärung
Accelerated	(= Beschleunigt). In der Accelerated-Betriebsart werden die Video-Ausgänge des GDP 82786 extern gemultiplext, wodurch sich eine Vervielfachung der Pixelfrequenz erreichen läßt. Der Plotprozessor berücksichtigt dies bei seinen Zeichenoperationen
Bank	auch RAM-Bank: Eine Anzahl von Speicherchips, die zusammen die Wortbreite desd Prozessors besitzen. Hier 16 Bit. Eine Bank besteht also z. B. aus 4 Chips 64 K × 4 oder 16 Chips 256 K × 1 usw.
Deskriptor	Beschreibendes Element. In einem Deskriptor werden charakteristische Eigenschaften von bestimmten Objekten festgehalten (z. B. vom Aufbau eines Fensters)
Interleaved	Abwechselndes Zugreifen. Dieser Begriff wurde schon früher im Zusammenhang mit abwechselndem Zugriff von Haupt-CPU und Graphikprozessor auf den Bildspeicher verwendet. Hier bedeutet er abwechselndes Zugreifen auf zwei (nebeneinanderliegende) RAM-Banks. Dadurch kann auf die eine RAM-Bank zugegriffen werden, während sich die andere noch in der Precharge-Phase befindet.
Precharge-Zeit/Phase	Beim Zugriff auf dynamische Speicher werden verschiedene Phasen durchlaufen. Die Precharge-Zeit ist eine Wartezeit, die nach einem Zugriff auf das DRAM eingehalten werden muß, damit das DRAM die Speicherzellen wieder neu aufladen kann.

Literatur

[1] „82786 CHMOS Graphics Coprocessor", Intel, Order Number 231676-001

[2] „The 82786 CHMOS Graphics Coprocessor Architectural Overview", Intel AP-259, Order Number 122711-001

[3] „82786 Hardware Configuration", Intel AP-270, Order Number 292007-001

Alle Datenblätter sind zu beziehen bei:
Intel Semiconductor
Seidlstraße 27
8000 München 2

Ulrich Rohde

Was Sie über Modula wissen sollten

Teil 3

In diesem Teil unserer Serie geht es an die eigentlichen Eigenschaften von Modula. Ausgehend vom Prinzip der Unterprogramme wird das Konzept der Module geschildert, das gegenüber Pascal große Vorteile aufweist und das auch das modernste Basic nicht besitzt.

Viel Kritik bekam ich zu hören, als Teil 1 draußen war. Ich würde kein modernes Basic kennen, hieß es, und meine Absicht, gegen Basic etwas zu sagen, würde dem Artikel aus allen Knopflöchern gucken. Gern gebe ich zu, daß das Beispiel GGT in Basic kurz war und übersichtlich, aber trotzdem ist an Modula etwas dran, was für alle fortschrittlich wäre, auch für die Freunde von Better Basic, Locomotive Basic2, Quick Basic und wie sie alle heißen mögen. Wer aber bei Basic bleiben will, der soll das von ganzem Herzen tun – und dem empfehle ich einen Artikel aus mc, Ausgabe 2, 1986, Seite 66. Was wurde in Teil 2 geboten? Erstens wurde festgestellt, daß man in Modula beinahe wie in Pascal programmieren kann. Es gibt viele Sprachelemente, die nahezu identisch sind. Auch die Technik, Prozeduren zu definieren, ist ähnlich, wirklich nur mit kleinen Unterschieden. Wenn in Modula eine Funktion definiert werden soll, dann ist das eine Prozedur, in deren Kopf nach den formalen Parametern noch eine Typangabe für das Ergebnis der Funktion folgt und innerhalb des Rumpfes der Funktionsprozedur muß wenigstens eine RETURN-Anweisung stehen, die das Ergebnis nach außen schleust. Der Prozedurname enthält nach dem Aufruf das Ergebnis.
Beispiel:

```
PROCEDURE invert(x:BOOLEAN):
                BOOLEAN;
BEGIN
RETURN NOT x;
END invert;
```

invert besitzt ein x vom Typ BOOLEAN als Parameter und will einen Wert vom Typ BOOLEAN ausgeben. Im Hauptprogramm müßte dann irgendwo eine Zuweisung a:=invert(f); stehen.

```
PROCEDURE Ackermann(a, b : CARDINAL) : CARDINAL;
  BEGIN
    IF a=0 THEN
      RETURN b+1;
    ELSE
      IF b=0 THEN
        RETURN Ackermann(a-1,1);
      ELSE
        RETURN Ackermann(a-1,Ackermann(a,b-1));
      END;
    END;
  END Ackermann;
```

Bild 1. Die Funktion Ackermann

Rekursion ist bei Prozeduren genauso wie in Pascal möglich. Tief rekursiv ist zum Beispiel die Ackermannfunktion, die hier nach dem Buch „Dal Cin/Lutz/Risse: Programmierung in Modula-2, Teubner Verlag, Stuttgart" programmiert ist *(Bild 1)*.
Unterprogramme sind in Modula im Programmtext als abgeschlossene Teile erkennbar. In der Vereinbarung einer Prozedur werden alle Aktionen, die zur Prozedur gehören, mit einem Namen verbunden. Mit diesem Namen läßt sich dann das Unterprogramm aufrufen. Unterprogramme sind zunächst das klassische Hilfsmittel, wenn komplexere Aufgaben zu programmieren sind. Man versucht das komplexe Problem in Teilaufgaben zu zerlegen und diese in einzelnen Prozeduren zu programmieren, um dann daraus ein Gesamtprogramm aufzubauen. Das macht große Programme übersichtlicher und auch ökonomischer in der Niederschrift, wenn ein und dieselbe Prozedur mehrmals aufgerufen wird. Die Schnittstellen der Unterprogramme zum Hauptprogramm, das sind die Parameter und bei Funktionsprozeduren dazu noch der Name der Prozedur selbst. Genauso, wie Prozeduren in Hauptprogrammen vereinbart werden können, ist es auch möglich, sie innerhalb von Prozeduren als „Unterprozeduren" zu vereinbaren. Die Tiefe der Schachtelung ist dabei praktisch unbegrenzt. *Bild 2* zeigt einen Modul, der invert und Ackermann zu einem Programm zusammenfaßt.

Parameter und Namen

Der Prozedurkopf gibt gewissermaßen bekannt, wie die Schnittstelle der Prozedur zur Umgebung aussehen soll. Während der Deklaration werden dort die formalen Parameter (in den Programmen aus Bild 1 und 2 sind sie x, y und z) genannt. Bei einem Aufruf müssen die aktuellen Parameter in der Reihenfolge der formalen Parameter genannt werden und im Typ mit diesen übereinstimmen. Das prüft der Compiler nach. Es sei noch darauf hingewiesen, daß schon im letzten Teil betont wurde, daß die formalen Parameter, die ohne VAR im Prozedurkopf genannt werden, bewirken, daß der Wert des genannten aktuellen Parameters beim Aufruf an die Prozedur übergeben wird. Der für den „Wertparameter" eingesetzte Ausdruck wird beim Aufruf ausgewertet. Der Ausdruck muß nicht nur eine Konstante oder Variable sein. Entsprechend wird auch Speicherplatz für die Prozedur angelegt. Mit VAR bei der Vereinbarung wird nur die Speicheradresse des aktuellen Parameters beim Aufruf an das Unterprogramm weitergegeben, und das Unterprogramm kann diesen Parameter an seinem Speicherort verändern. Weil VAR-Parameter berechnete Ergebnisse aus der Prozedur heraustransportieren können, und das auch sollen, dürfen an ihrer Stelle nur Variablen eingesetzt werden. Der neue Inhalt bleibt nach Ende des Unterprogrammes bestehen, der alte ist verloren. Es gibt eine weitere Art der Para-

meterübergabe. Bei Arrays weiß man beim Anlegen eines Unterprogrammes oft nicht, wieviele Komponenten es besitzt. Deshalb kann man offene ARRAY-Typen übergeben. Beispiel eines Prozedurkopfes für eine Prozedur, die Botschaften verwursten soll:
PROCEDURE message(Botschaft: ARRAY OF CHAR);
Die Namen der formalen Parameter aller Arten sind nur für das Innere der Prozedur sichtbar. Von außen kann derselbe Name ohne Konflikt als aktueller Parameter eingesetzt werden, wie im Beispiel aus Bild 2 gezeigt wurde. In einer Prozedur sind zunächst alle global vereinbarten Namen sichtbar und können benutzt werden. Wird aber in einer Prozedur ein global vereinbarter Name nochmals neu vereinbart, so schattet die lokale Version den übergeordneten Namen ab. Im Unterprogramm wird nur das lokale Exemplar benutzt. Man könnte also aus einem Unterprogramm heraus alles Sichtbare aus der Umgebung benutzen. Guter Programmierstil ist das aber nicht. In einem Unterprogramm sollen nur Namen benutzt werden, die entweder über die Liste der formalen Parameter angegeben sind oder die lokal vereinbart sind. Das hilft gegen Namenskonflikte und Nebeneffekte (unbeabsichtigte und schwer zu entdeckende Veränderungen an Variablen). Alles, was innerhalb eines Unterprogrammes an lokalen Namen vereinbart ist, wird erst ins Leben gerufen, wenn die Prozedur aufgerufen ist. Nach ihrem Ende sind alle lokalen Namen vergessen und der damit zusammenhängende Speicher ebenfalls. Im vorigen Teil gab es eine Liste, in der global definierte Prozeduren aufgeführt waren, die als überall vereinbart gelten.

Module

Was in Pascal Programm heißt, das heißt in Modula Hauptmodul. Ein Hauptmodul besteht, so wurde es im letzten Teil gezeigt, aus dem Kopf mit Namen und einer fakultativ anzugebenden Prioritäts-Kennzahl, die sein Verhalten gegenüber parallel laufendem Geschehen im System steuert und die hier nur erwähnt sei. Danach folgen Importlisten und dann der Block des Moduls mit den Vereinbarungen und dem eigentlichen Anweisungsteil nach dem BEGIN. In den Importlisten ist angegeben, welche anderen Module und oft auch, welche Objekte er aus anderen Modulen importiert. In Bild 2 wird der Modul InOut importiert. Danach können weiter unten in Bild 2 die Prozeduren InOut.WriteString(...) usw. benutzt werden. Kurz gesagt, enthält InOut eine Anzahl nützlicher Routinen, die durch den Import von InOut dem Modul in Bild 2

```
MODULE Test;
  IMPORT InOut;
  VAR
    x : BOOLEAN;
    y, z : CARDINAL;
  PROCEDURE invert(x : BOOLEAN) : BOOLEAN;
    BEGIN
      RETURN NOT x;
    END invert;
  PROCEDURE Ackermann(a, b : CARDINAL) : CARDINAL;
    BEGIN
      IF a=0 THEN
        RETURN b+1;
      ELSE
        IF b=0 THEN
          RETURN Ackermann(a-1,1);
        ELSE
          RETURN Ackermann(a-1,Ackermann(a,b-1));
        END;
      END;
    END Ackermann;
  BEGIN
      (*Hauptprogramm*)
    InOut.WriteString("bitte y eingeben");
    InOut.WriteLn();
    InOut.ReadCard(y);
    InOut.WriteLn();
    InOut.WriteString("bitte z eingeben");
    InOut.WriteLn();
    InOut.ReadCard(z);
    InOut.WriteLn();
    InOut.WriteCard(Ackermann(y,z), 6);
    InOut.WriteLn();
    IF invert(FALSE) THEN
      InOut.WriteString("FALSE ist invertiert!");
    END;
  END Test.
```

Bild 2. Ein Hauptmodul, der invert und Ackermann enthält und aufruft

```
MODULE Modtest2;
  IMPORT InOut;
  VAR
    x : BOOLEAN;
    y, z : CARDINAL;
  MODULE Lokal;
    EXPORT Ackermann, invert;
    PROCEDURE invert(x : BOOLEAN) : BOOLEAN;
      BEGIN
        RETURN NOT x;
      END invert;
    PROCEDURE Ackermann(a, b : CARDINAL) : CARDINAL;
      BEGIN
        IF a=0 THEN
          RETURN b+1;
        ELSE
          IF b=0 THEN
            RETURN Ackermann(a-1,1);
          ELSE
            RETURN Ackermann(a-1,Ackermann(a,b-1));
          END;
        END;
      END Ackermann;
    BEGIN
    END Lokal;
  BEGIN
      (*Hauptprogramm*)
    InOut.WriteString("bitte y eingeben");
    InOut.WriteLn();
    InOut.ReadCard(y);
    InOut.WriteLn();
    InOut.WriteString("bitte z eingeben");
    InOut.WriteLn();
    InOut.ReadCard(z);
    InOut.WriteLn();
    InOut.WriteCard(Ackermann(y,z), 6);
    InOut.WriteLn();
    IF invert(FALSE) THEN
      InOut.WriteString("FALSE ist invertiert!");
    END;
  END Modtest2.
```

Bild 3. Hier sind invert und Ackermann in einem lokalen Modul verpackt

zur Verfügung stehen. Die Prozeduren aus InOut leisten Hilfestellung bei der Kommunikation mit der Peripherie. Das Besondere hier ist die Ansprache der Unterprogramme aus InOut. Dazu werden die Namen der Prozeduren aus InOut mit dem vorgesetzten und durch Punkt abgetrennten Namen des Moduls, aus dem sie kommen, ergänzt. Diese Qualifizierung des Namens hilft Namenskonflikte vermeiden.

Im Hauptmodulblock können wiederum lokale Module vereinbart sein. Ein Modula-Programm stellt deshalb in der Regel eine Hierarchie aus mehreren Modulen dar. Das Konzept der Module ist gegenüber dem Konzept der Unterprogramme wesentlich reicher in den Gestaltungsmöglichkeiten. Ein Modul stellt einem anderen, der ihn nutzt, kontrolliert seine Fähigkeiten (Prozeduren) und seine Datenobjekte zur Verfügung.

Das ist gut an folgendem Beispiel *(Bild 3)* zu sehen, in dem die beiden Unterprogramme invert und Ackermann in einen lokalen Modul aufgenommen sind. Das hat hier keinen weiteren Sinn, als die formale Übung. Das Programm ist funktionsgleich zu dem aus Bild 2, aber man kann sehen, daß die beiden Prozeduren jetzt exportiert werden müssen, damit sie außerhalb des Moduls benutzt werden können. In einem Modul können also Objekte, hier Unterprogramme, zusammengefaßt sein, die von anderer Stelle aus benutzt werden sollen. Der tiefere Sinn solcher Verpackung von Objekten in lokale Module erschließt sich erst, wenn man komplexere Berechnungen vor hat, deren Details im übergeordneten Programm verborgen bleiben soll – oder wenn man komplexere Datenstrukturen benötigt, die von vielen Stellen aus genutzt werden sollen – ohne daß die Details ihrer Implementation überall bekannt sein müssen. Solche komplexen Beispiele sollen hier noch nicht verwendet werden, weil sie den Blick auf die Technik der Programmierung in Modula zunächst verstellen würden. Das einfache Beispiel soll noch etwas weiter gequält werden. Im nächsten Schritt bekommt der Modul Lokal selbst noch einen Anweisungsteil. Außerdem eine Importliste.

Der Modul Lokal importiert einerseits die Variablen x,y,z aus einem übergeordneten Modul, damit er die Prozeduren invert und Ackermann auch selbst nutzen kann, wie es sein Anweisungsteil vorsieht. Andererseits importiert er von InOut bestimmte Objekte, nämlich ReadCard, WriteCard und so weiter. Bei dieser Art des Importes muß der qualifizierende Zusatz, der die Herkunft der Objekte vor dem Namen bezeichnet, später bei Aufruf eines Objektes nicht genannt

```
MODULE Modtest3;
  IMPORT InOut;
  VAR
    x : BOOLEAN;
    y, z : CARDINAL;
  MODULE Lokal;
    IMPORT x, y, z;
    FROM InOut IMPORT ReadCard, WriteCard, WriteLn, WriteString;
    EXPORT Ackermann, invert;
    PROCEDURE invert(x : BOOLEAN) : BOOLEAN;
      BEGIN
        RETURN NOT x;
      END invert;
    PROCEDURE Ackermann(a, b : CARDINAL) : CARDINAL;
      BEGIN
        IF a=0 THEN
          RETURN b+1;
        ELSE
          IF b=0 THEN
            RETURN Ackermann(a-1,1);
          ELSE
            RETURN Ackermann(a-1,Ackermann(a,b-1));
          END;
        END;
      END Ackermann;
    BEGIN
      WriteString("bitte y eingeben");
      WriteLn();
      ReadCard(y);
      WriteLn();
      WriteString("bitte z eingeben");
      WriteLn();
      ReadCard(z);
      WriteLn();
      WriteCard(Ackermann(y,z), 6);
      WriteLn();
    END Lokal;
  BEGIN
      (*Hauptprogramm*)
    InOut.WriteString("bitte y eingeben");
    InOut.WriteLn();
    InOut.ReadCard(y);
    InOut.WriteLn();
    InOut.WriteString("bitte z eingeben");
    InOut.WriteLn();
    InOut.ReadCard(z);
    InOut.WriteLn();
    InOut.WriteCard(Ackermann(y,z), 6);
    InOut.WriteLn();
    IF invert(FALSE) THEN
      InOut.WriteString("FALSE ist invertiert!");
    END;
  END Modtest3.
```

Bild 4. Mit Importen können invert und Ackermann auch im lokalen Modul angesprochen werden. Anweisungsteile in Modulen dienen der Initialisierung, wenn es kein Hauptmodul ist

```
DEFINITION MODULE InOut;
 (*************************************************************************
  *    Copyright (C) 1984, 85 Interface Technologies Corporation.         *
  *    All Rights Reserved.                                               *
  *                                                                       *
  *    Registered users are hereby granted the right to use the           *
  *    following program in custom applications.                          *
  *    This permission does not include the right to redistribute         *
  *    this code.                                                         *
  *************************************************************************)
 (*************************************************************************
  * Purpose: InOut is a stream I/O package that allows the user to do
  *          I/O on files, terminals, and other devices in a consistent
  *          fashion. This implementation is based on the one presented
  *          in "Programming in Modula-2" (2nd Ed.) by Niklaus Wirth.
  * Date   : 11-84
  * Version: 1.1
  *************************************************************************)
 FROM SYSTEM IMPORT ADDRESS, STRING;
 EXPORT QUALIFIED Done, in, out, termCH, OpenInput, OpenOutput,
     CloseInput, CloseOutput, Read, ReadAOC, ReadString, ReadInt,
     ReadCard, ReadRec, Write, WriteAOC, WriteLn, WriteString, WriteInt,
     WriteCard, WriteOct, WriteHex, WriteRec;
```

Bild 5. Der DEFINITION-Modul von InOut des M2SDS-Compilers von Interface Technologies

```
TYPE
  Stream;
VAR
  termCH : CHAR;
                        (* Set by 'ReadString', 'ReadInt', 'ReadCard';
                         * it is the charcter that actually terminated
                         * the input: EOL, esc, space, nul etc.
                         *)
  Done : BOOLEAN;
                        (* 'Done' is set by several procedures; TRUE if the
                         * operation was successful, FALSE otherwise.
                         *)
  in, out : Stream;
                        (* The currently open input and output files. *)
PROCEDURE OpenInput(fname : STRING);
  (* Opens input stream 'fname'. If 'fname' is prefixed by a drive
   * specification, that drive is used, otherwise the currently selected
   * drive is used. If the file spec. in 'fname' begins with 'Ö', 'fname'
   * is assumed to be an absolute path name. If the file spec. does not
   * begin with 'Ö', 'fname' is assumed to be a suffix for the current
   * directory which ends in 'Ö'. If 'OpenInput' succeeds, 'Done' = TRUE,
   * and subsequent input is taken from the stream until 'CloseInput' is
   * called.
   *)
PROCEDURE OpenOutput(fname : STRING);
  (* Opens output stream 'fname'. Drive and path name conventions are the
   * same as described under 'OpenInput'.
   *)
PROCEDURE CloseInput();
  (*.Closes the current input file. *)
PROCEDURE CloseOutput();
  (* Closes the current output file. *)
PROCEDURE Read(VAR ch : CHAR);
  (* Reads the next character from the current input stream. 'Done' = TRUE
   * unless the input is at end of file.
   *)
PROCEDURE ReadAOC(VAR a : ARRAY OF CHAR);
  (* Reads characters from the current input stream into 'a' until an EOL
   * is detected. 'ReadAOC' truncates the input if it is too long for 'a'.
   *)
PROCEDURE ReadString(VAR s : STRING);
  (* Reads a string from the current input stream. Characters are read
   * into 's' until an EOL is detected. 'ReadString' truncates the input
   * string if it is too long for 's'.
   *)
PROCEDURE ReadInt(VAR x : INTEGER);
  (* Reads an INTEGER number from the current input stream. 'Done' = TRUE
   * if what was read converted correctly to an INTEGER.
   *)
PROCEDURE ReadCard(VAR x : CARDINAL);
  (* Reads a CARDINAL number from the current input stream. *)
PROCEDURE ReadRec(r : ADDRESS; size : CARDINAL) : CARDINAL;
  (* Reads the next 'size' bytes into a buffer at 'r'. 'r' is assumed to
   * be large enough to hold 'n' bytes. Returns the number of bytes
   * actually read. If there is an I/O error, or if end of file is
   * reached, this will be less than 'size'.
   *)
PROCEDURE Write(ch : CHAR);
  (* Writes a character to the current output stream. *)
PROCEDURE WriteAOC(a : ARRAY OF CHAR);
  (* Writes characters from 'a' to the current output stream. Output is
   * terminated by an EOL, nul, or the end of the character array,
   * whichever comes first.
   *)
PROCEDURE WriteLn();
  (* Writes an EOL to the current output stream. *)
PROCEDURE WriteString(s : STRING);
  (* Writes a string to the current output stream. *)
PROCEDURE WriteInt(x : INTEGER; n : CARDINAL);
  (* Writes an INTEGER number, right-justified, to the current output
   * stream. At least 'n' characters are used (more if required).
   *)
PROCEDURE WriteCard(x, n : CARDINAL);
  (* Writes a CARDINAL number, right-justified, to the current output
   * stream. At least 'n' characters are used (more if required).
   *)
PROCEDURE WriteOct(x, n : CARDINAL);
  (* Writes a CARDINAL number in octal format to the current output stream.
   * At least 'n' characters are used (more if required).
   *)
PROCEDURE WriteHex(x, n : CARDINAL);
  (* Writes a CARDINAL number in hex format to the current output stream.
   * At least 'n' characters are used (more if required).
   *)
PROCEDURE WriteRec(r : ADDRESS; size : CARDINAL);
  (* Appends the record pointed to by 'r' to the stream. The record is
   * 'size' bytes in length.
   *)
END InOut.
```

werden. Jetzt kommen weiter Unterschiede zu Unterprogrammen zum Tragen. Wenn ein Modul, wie in Bild 4, einen Anweisungsteil besitzt, dann wird dieser abgearbeitet. Entweder ist das nämlich ein Anweisungsteil eines Programmodules und muß als Hauptprogramm absolviert werden oder ein Anweisungsteil eines lokalen oder importierten Moduls, dann muß er als Initialisierungsteil dieses Moduls ablaufen – und zwar vor dem Anweisungsteil des Haupmodules. Die Anweisungsteile der Module werden in der Reihenfolge der Deklaration absolviert. In Bild 4 wird also erst einmal Ackermann bei der Initialisierung des Moduls ausgeführt und dann werden wiederum Ackermann und invert in dem Hauptmodul aufgerufen.

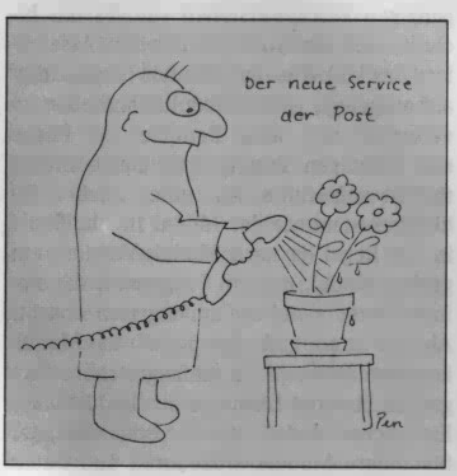
Jetzt kann im Sinn von Niklaus Wirth gesagt werden: Ein Modul ist eine Zusammenfassung von Datenstrukturen und Prozeduren für eine bestimmte Aufgabe. Die Import- und Exportlisten sind ein Werkzeug, um gezielt und kontrolliert Objekte für andere zugänglich zu machen oder auch verborgen zu halten. Die Objekte in den Modulen existieren statisch. Im Unterschied zu Unterprogrammen wird der Anweisungsteil eines Modules nur einmal zur Initialisierung abgearbeitet, er ist von außen nicht zugänglich.

Externe Module

Was bis jetzt über lokale Module gesagt wurde, sollte vor allem auch darauf vorbereiten, was externe Module in Modula bedeuten. Jeder Programmierer weiß, daß er schon vieles in speziellen Aufgaben programmiert hat, was so allgemeingültig ist, daß es in sehr viel mehr Programmen eingesetzt werden kann, als nur dem einen, in dem die Aufgabe zum ersten Mal gelöst wurde. Assembler und viele Hochsprachen besitzen daher Bibliotheksfunktionen, die es erlauben, vorproduzierte Programme in neu zu schreibende einzubinden. Bei den gängigen Pascal-Versionen sind es das Include- und Unitkonzept, bei den Assemblern das Link-Konzept. Modula ist ganz darauf ausgelegt, ganze Modulbibliotheken zu verwalten und seine Benutzer zur Pflege und ständigen Erweiterung dieser Bibliotheken aufzurufen. Aus einer solchen Bibliothek stammte der Modul InOut. Es ist in der Regel (je nach Modula-Version) ein großer Modul, der ein Programm, das ihn importiert, ziemlich umfangreich macht. Aber er ist ein gutes Beispiel für das Modulkonzept. Denn jedes umfangreichere Programm benötigt Elemente aus InOut.

Ein solcher Modul aus einer Bibliothek, der für andere Module Prozeduren und Daten

bereit hält, besteht aus zwei Teilen. Einmal aus der Schnittstelle für seine Kunden, in der exakt angegeben ist, was er für Elemente enthält, von wo er selbst etwas importiert und was er für seine Kunden öffentlich bereit hält. Dieser Teil heißt DEFINITION-Modul. Der Beginn des DEFINITION-Moduls aus der Modula-Version M2SDS ist in *Bild 5* gezeigt. *Bild 6* zeigt einen Ausschnitt aus dem zweiten Teil dieses Moduls, den IMPLEMENTATION-Modul. Dort sind die Prozeduren ausprogrammiert, die die im DEFINITION-Modul definierten Funktionen auf dem Rechner implementieren. Beide Module gehören zusammen, können aber im Modula-System getrennt behandelt werden und auch separat compiliert werden. Der Sinn der Sache ist folgender: Unabhängig von der Realisierung einer Funktion steht ja fest, was sie formal zu leisten hat. Das kann in Form einer Definition – hier eines Prozedurkopfes – niedergelegt werden. Damit ist bekannt, welche Parameter ein Benutzer anliefern muß, um aus der Prozedur die gewünschten Ergebnisse zu erhalten. Die eigentliche Implementation muß ebenfalls nur auf die Schnittstelle achten, was sie an Parametern zu erwarten hat und was sie daraus zu machen hat. Sie kann gegebenenfalls geändert werden, vielleicht, weil eine neue Idee Speicherplatz sparen hilft oder Tempo gewinnen. Wenn nun ein Modul in Modula compiliert werden soll, der einen anderen Modul oder Objekte daraus importiert, dann schaut der Compiler zunächst im DEFINITION-Modul nach, ob es auch seine Richtigkeit mit der Verwendung der importierten Objekte hat. Wenn das alles paßt, dann kann im weiteren Durchgang alles zusammengelinkt werden, so daß ein lauffähiges Modula-Programm entsteht. Soweit zunächst an kleinen Beispielen, was es mit den Modulen auf sich hat. Im nächsten Teil geht es noch mehr um die Beziehungen von Modulen untereinander. *Wird fortgesetzt*




```
IMPLEMENTATION MODULE InOut;
  (*************************************************************************
   *    Copyright (C) 1984, 85 Interface Technologies Corporation.         *
   *    All Rights Reserved.                                               *
   *                                                                       *
   *    Registered users are hereby granted the right to use the           *
   *    following program in custom applications.                          *
   *    This permission does not include the right to redistribute         *
   *    this code.                                                         *
   *************************************************************************)
  (*************************************************************************
   * Purpose: InOut is a stream I/O package that allows the user to do
   *          I/O on files, terminals, and other devices in a consistent
   *          fashion. This implementation is based on the one presented
   *          in "Programming in Modula-2" (2nd Ed.) by Niklaus Wirth.
   * Author :          M.K. Donegan
   * Date   : 11-84    12-85
   * Version: 1.1      1.2
   *************************************************************************)
  FROM SYSTEM IMPORT ADDRESS, WORD, ADR, STRING;
  FROM ASCII IMPORT space, cr, esc, bs, EOL;
  IMPORT NumConv;
  IMPORT Files;
  TYPE
    Stream = Files.File;
  VAR
    termin, termout : Stream;
    termIsInput, termIsOutput : BOOLEAN;
  PROCEDURE OpenInput(fname : STRING);
    BEGIN
      IF (fname[1]='T') AND (fname[2]=':') THEN
        in := termin;
        termIsInput := TRUE;
      ELSE
        in := Files.Open(fname,Files.Rmode);
        termIsInput := FALSE;
        IF Files.status#Files.Done THEN
          in := termin;
          termIsInput := TRUE;
        END;
      END;
      Done := (Files.status=Files.Done);
    END OpenInput;
  PROCEDURE OpenOutput(fname : STRING);
    BEGIN
      IF (fname[1]='T') AND (fname[2]=':') THEN
        out := termout;
        termIsOutput := TRUE;
      ELSE
        out := Files.Create(fname,Files.Wmode);
        termIsOutput := FALSE;
        IF Files.status#Files.Done THEN
          out := termout;
          termIsOutput := TRUE;
        END;
      END;
      Done := (Files.status=Files.Done);
    END OpenOutput;
  PROCEDURE CloseInput();
    BEGIN
      IF NOT termIsInput THEN
        Files.Close(in);
        in := termin;
        termIsInput := TRUE;
      END;
    END CloseInput;
  PROCEDURE CloseOutput();
    BEGIN
      IF NOT termIsOutput THEN
        Files.Close(out);
        out := termout;
        termIsOutput := TRUE;
      END;
    END CloseOutput;
  PROCEDURE Read(VAR ch : CHAR);
    BEGIN
      IF termIsInput THEN
        REPEAT
          Files.ReadChar(in, ch);
        UNTIL (Files.status=Files.Done);
        Done := TRUE;
      ELSE
        Files.ReadChar(in, ch);
        IF Files.status=Files.Done THEN
          Done := TRUE;
        ELSE
          Done := FALSE;
          ch := 0C;
```

Bild 6. Ausschnitt aus dem dazugehörenden IMPLEMENTATION-Modul

SOFTWARE

ODER

SERVICE

WIR BIETEN BEIDES !

Die Bibliothek der Superlative für große, netzwerkfähige Applikationen unter MS-DOS heißt Modula-2 Advanced Library/Net und kostet 4500,– DM. Nur einige Stichworte: Netzwerkfähige ISAM mit File- und Record-Locking, Datenbankbrowser, Masken- und Menügenerator, Hilfefenster, Makros, Grafiktreiber, Druckformulare, Modem-Unterstützung, Desk Accessories. All dies selbstverständlich im Source-Code und frei von Run-Time-Lizenzen. Mit dieser Umgebung können Sie getrost auf programmierbare Datenbanken und 4GLs verzichten – Sie bekommen deren Mächtigkeit, büßen aber nicht die Vorteile einer modernen, modularen Sprache ein. Nähere Informationen und Demo-Diskette auf Wunsch. Ferner im Angebot: Compiler für alle wichtigen Systeme, und Bibliotheken zu allen Anwendungsgebieten. Fordern Sie unseren Katalog an!

Wir haben mehr als 60 Toolboxes für Turbo Pascal zu den Stichworten: 8087, BCD, BIOS-Calls, Basic-Konverter, Bildschirmsteuerung, DOS-Calls, Debugger, Desk Accessories, Disassembler, Disk Cache, Echtzeitanwendungen, Editor, Expanded Memory, Extended Memory, FFT, Formulargenerator, GEM, Grafik, Hochauflösende Karten, ISAM, Kommunikation, Kurvenanpassung, Lernprogramme, Linker, Makros, Maskengenerator, Menüs, Multi-Tasking, Netzwerk, Numerik, Optimierung, Optimizer, Plotter, Profiler, Programmgenerator, RS232, SQL, Simulation, Sortieren, Statistik, TSRs, Zeichensätze. Weitere Pascal-Compiler für MS-DOS, 80386, UNIX, VMS u.a. Detaillierte Beschreibungen in unserem Katalog.

LOGITECH

L&W

mts

WILKEN & SABELBERG Software GbR

ENZ EDV-BERATUNG GMBH

Turbo Pascal ist eingetragenes Warenzeichen von Borland Int.

Werner Hilf

Der MC68030

Teil 1: Aufbau und Hardware

Mit einer neuen Architektur, einem höheren Integrationsgrad und einem verbesserten Chip-Layout hat sich Motorola einen weiteren Vorsprung auf dem 32-Bit-Sektor verschafft: Welche Möglichkeiten und Funktionen der Prozessor 68030 bietet, wird in dieser Artikel-Serie dargestellt.

Immer mehr Anwender verlangen nach höher integrierten Bausteinen und schnelleren Mikroprozessoren. Ein großer Schritt in diese Richtung ist der MC68030, der seit Anfang 1988 von der Muster- in die Produktionsphase übergegangen ist – Grund genug, die Funktion dieses modernen Prozessors zu beschreiben.

Einführung

Was sind die entscheidenden Merkmale des 68030, der auch „IP" (integrierter Prozessor) genannt wird? Hier eine Auflistung:

- voll kompatibel zum MC68020
- Harvard-Architektur
- 20 MHz Standard-Taktfrequenz
- integrierte PMMU (Teilmenge des MC68851)
- 256 Byte Befehlscache
- 256 Byte Datencache
- verbesserter Buscontroller
- maximal 2,5 Watt Verlustleistung
- gefertigt in der 1,2µ-HCMOS-Technik

In diesem Artikel wird hauptsächlich auf die Unterschiede zum Prozessor der sogenannten „ersten 32-Bit-Generation", dem MC68020, eingegangen. Er ist auf diesem Sektor marktbeherrschend und erfreut sich bereits großer Beliebtheit auch im PC-Bereich. Er wird sowohl im klassischen 68000-Anwendungsbereich, der Steuerungs- und Regelungstechnik, als auch in Grafikapplikationen eingesetzt. Führend ist seine Rolle ebenfalls in Unix-Mikroprozessor-Systemen. Letzteres hat seinen Grund in der Sprache C, deren Compiler mit Unterstützung des Programmiermodells, den Adressierungsarten und dem Befehlssatz der 68000-Familie effizient geschrieben werden können.
Für den einfachen und sicheren Aufbau von Mehrplatz-Systemen ist eine Speicherverwaltungseinheit (MMU) notwendig, die eine Kollision der verschiedenen Anwender bzw. Programme im System verhindert. Damit erhöht sich jedoch der Hardware-Aufwand. Daher strebt man beim Einsatz von Mikroprozessoren in diesen Systemen eine Integration der MMU an. Beim 68030 wurde erstmals der größte Teil der MMU auf dem Chip integriert, die beim 68020 extern als Coprozessor (MC68851) angeschlossen werden mußte. Die daraus resultierende niedrige Verlustleistung und vor allem der geringe Platzbedarf führt zur Reduzierung der Systemkosten.
Systemkosten sind außerdem abhängig von der Leistungsfähigkeit der CPU, die ihrerseits von den Zugriffszeiten der Speicher mitbestimmt wird. Ein guter Kompromiß zwischen Kosten und Leistungsfähigkeit sind Caches. Dies sind sehr schnelle Speicher, die immer parallel zum Hauptspeicher liegen und vom Prozessor selbst geladen werden. Befehle oder Daten, die im Cache stehen, können mit maximaler Geschwindigkeit verarbeitet werden. Der 68030 verfügt über einen 256 Byte großen Befehls-Cache. Der 256 Byte fassende Daten-Cache wurde neu eingeführt, um dynamische RAMs besser zu unterstützen: So kann der Datencache im sogenannten „Burst-Fill-Modus" beschrieben werden. Wir werden darauf noch näher eingehen.
Neben der Lösung von Platzproblemen wünschen sich die Anwender immer wieder die Erhöhung der Geschwindigkeit. Harvard-Architektur, ein modifizierter Buscontroller und neue Halbleiter-Technologien sorgen beim 68030 für einen höheren Durchsatz. Er ist damit ungefähr zweimal so schnell wie der 68020.

Interner Aufbau

Computer bedienen sich bei der Abarbeitung eines Befehls üblicherweise des sogenannten „von-Neumann-Zyklus", d. h. Holen, Dekodieren und Ausführen einer Instruktion laufen sequentiell ab. Auf dem Mikroprozessor-Sektor ist die 68000-Familie bislang ein Vorbild: Immer wieder schafften es die Entwickler, die Prozessoren mit den Rechnerarchitekturen auszustatten, die bislang die Domäne der Großcomputer waren. Beim MC68000 [1] hatte der Mikrocode auf dem Chip ebenso Premiere wie der „Prefetch-Mechanismus". Letzterer dient der Erhöhung des Durchsatzes, indem man etwas vom „von-Neumann-Zyklus" abweicht: immer dann, wenn der Prozessor zur Ausführung den externen Bus nicht mehr benötigt, kann er – noch während er einen Befehl ausführt – schon den nächsten holen. Für diese Überlappung

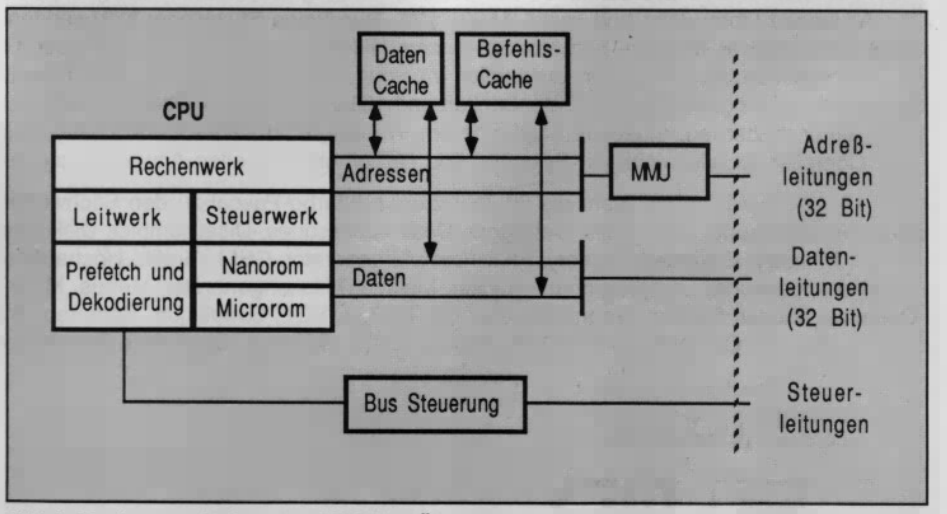


Bild 1. Der interne Aufbau des 68030 im Überblick

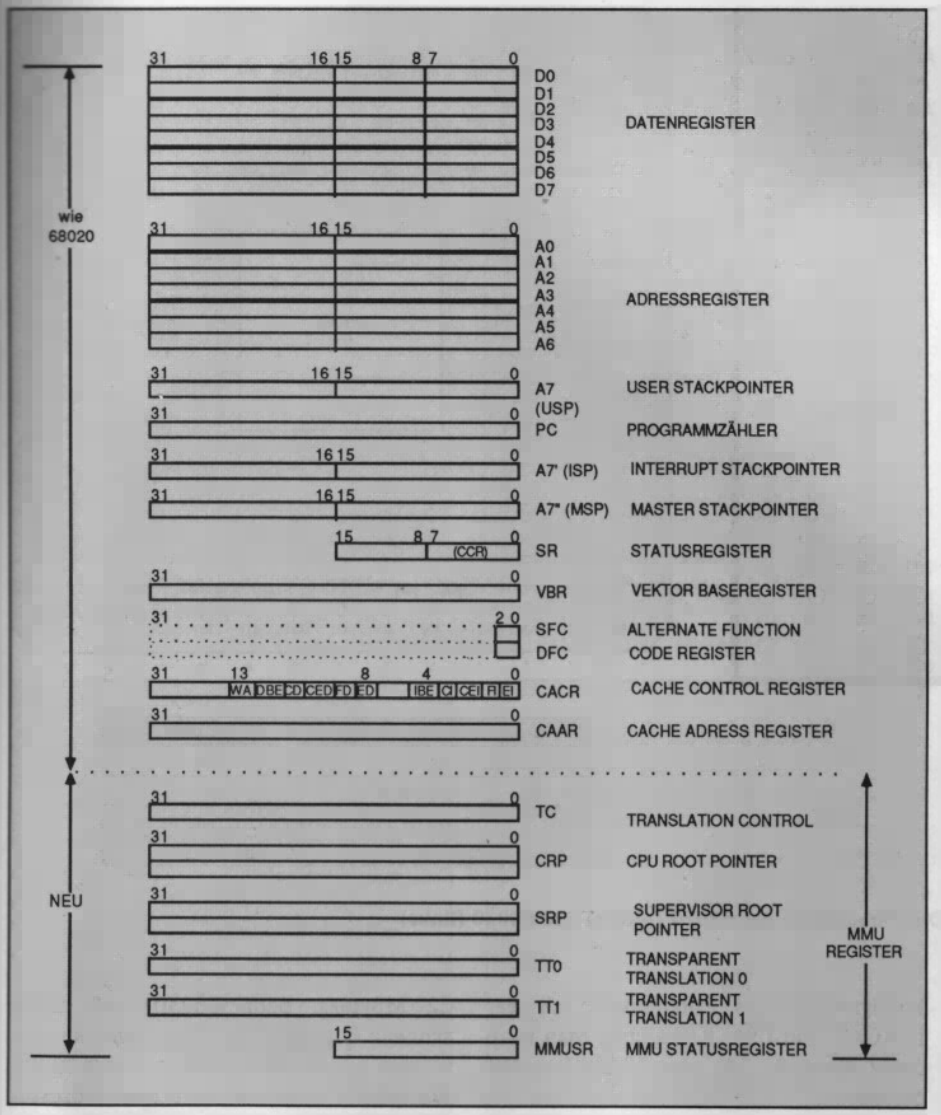


Bild 2. Im Registersatz des 68030 finden sich nun auch die MMU-Register

benötigt der Prozessor eine sogenannte Pipeline. Die komplette 32-Bit-Struktur, erweiterte Pipeline und ein Befehlscache auf dem Chip ermöglichen es dem 68020 [2], den parallelen Ablauf so zu steuern, daß Befehle gleichzeitig abgearbeitet werden können.

Bei den CPUs herkömmlicher Art werden Befehle und Daten intern über gemeinsame Adreß- bzw. Datenpfade geschickt. Eine Trennung bringt einen weiteren Geschwindigkeitsvorteil mit sich, da Befehle und Daten intern gleichzeitig und unabhängig voneinander verarbeitet werden können. Dieses Prinzip der getrennten Busse wird Harvard-Architektur genannt. In *Bild 1* ist dieser Aufbau zu erkennen.

Schematisch sind hier unter anderem die beiden Caches dargestellt. Sie liegen beide vor der MMU, sind also der CPU zugewandt und werden daher als logische Caches bezeichnet. Im Gegensatz dazu sind physikalische Caches dem physikalischen Speicher zugewandt.

Betrachten wir den Registersatz des 68030 *(Bild 2)*, so wird die Kompatibilität zum 68020 deutlich: Lediglich das Cache Control Register (CACR) wurde um Bits erweitert, die eine Manipulation des Datencaches in der gleichen Weise wie beim Befehlscache erlaubt. Die Register der MMU sind nun Bestandteil der CPU.

Hardware

Bild 3a ruft noch einmal die Signale des 68020 in Erinnerung. Signale, die neu gegenüber dem 68000 eingeführt wurden, sind mit einem Sternchen „*“ markiert [3].

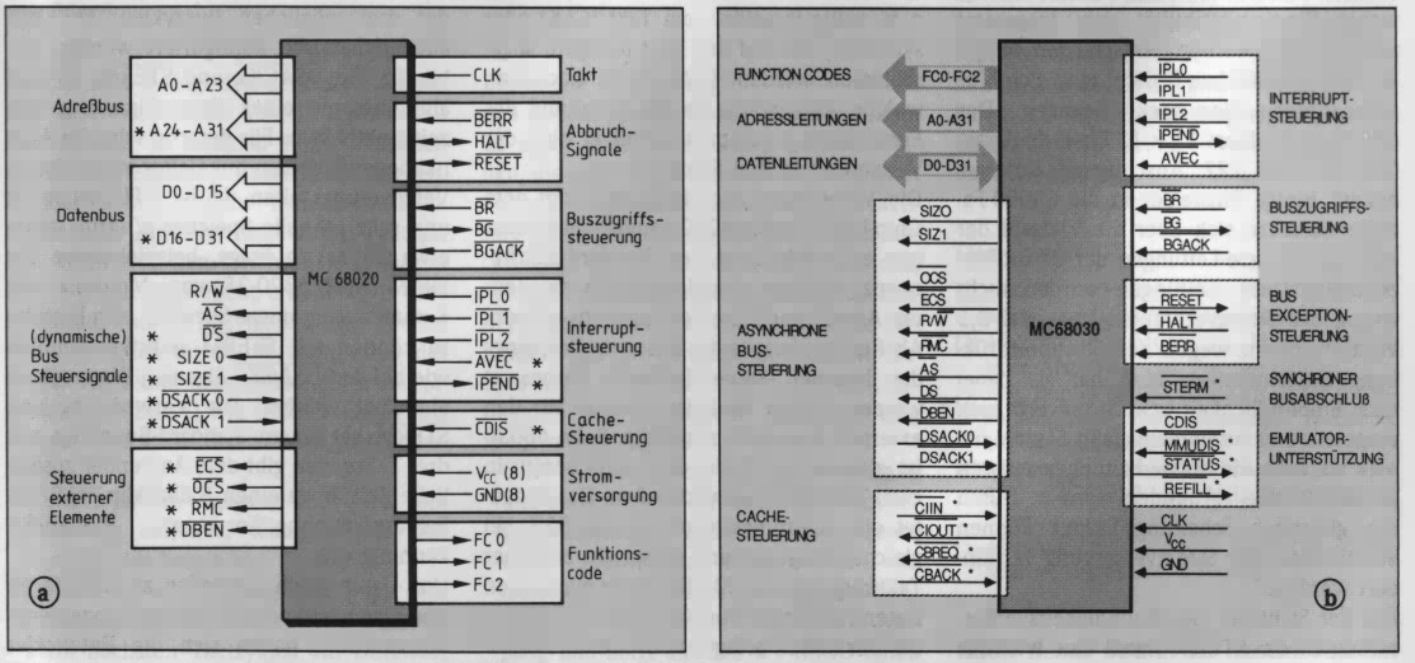
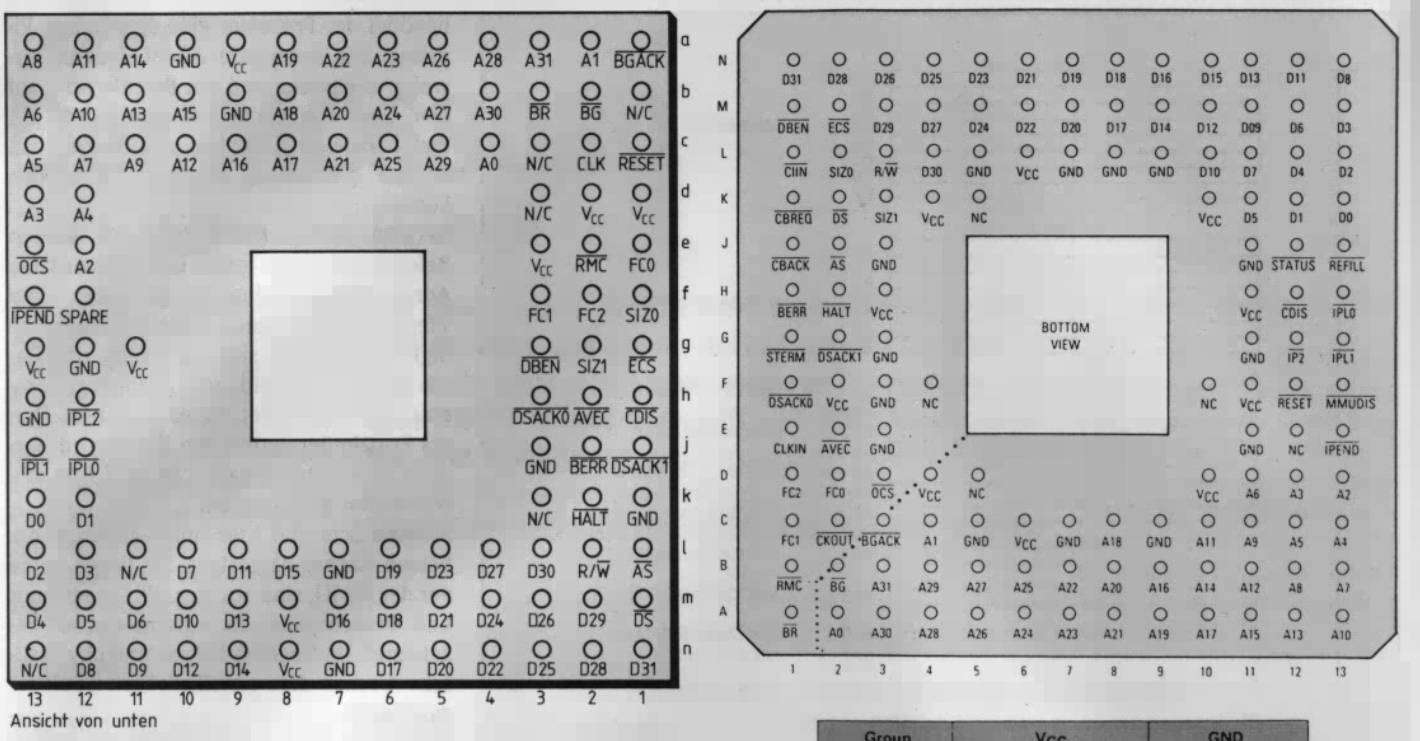
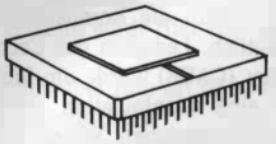


Bild 3. Mit diesen Signalen nehmen der 68020 (a) und der 68030 (b) die Verbindung zur Außenwelt auf





Seitenansicht

Group	V_{CC}	GND
Address Bus	A9, D3	A10, B9, C3, F12
Data Bus	M8, N8, N13	L7, L11, N7, K3
Logic	D1, D2, E3, G11, G13	G12, H13, J3, K1
Clock		B1

Bild 4. Die Pinbelegung des 68020 (rechts) und 68030 (links)

Der 68030 hat einige neue Signale gegenüber dem 68020 bekommen, sie sind in Bild 3b ebenfalls mit einem Stern versehen. Die drei Gruppen synchroner Bus, Emulator-Unterstützung und Cache-Steuerung wurden ergänzt bzw. neu hinzugefügt. Trotz der geringen Anzahl neuer Pins liegt wenig Pin-Kompatibilität vor; insgesamt zwölf Leitungen sind identisch (*Bild 4a* und *4b*). Zu Testzwecken bedient man sich hier sogenannter „Piggy-back"-Lösungen. Der 68030 ist in einem 13 × 13 Pin-Grid-Array-Gehäuse mit 122 Anschlüssen untergebracht. Einige Einsteiger in die 68000-Familie wundern sich über die Vielzahl der +5V- und Masse-Leitungen der 32-Bit-Prozessoren. Diese Leitungen wurden nicht wegen des Stromverbrauchs (maximal 2,5 Watt), sondern wegen der Störunterdrükkung implementiert. Man hat sie dabei nach Empfindlichkeit bzw. Stromverbrauch eingeteilt und zusammengefaßt. Sowohl die +5V als auch die Masse-Leitungen müssen auf der Platine verbunden werden. Durch ein günstiges Schaltungs-Layout können Störeinflüsse der Stromversorgung vermindert werden.

Aus der Sicht der Geschwindigkeitsverbesserung ist das $\overline{\text{STERM}}$-Signal eine wichtige Leitung. Der 68020 erlaubt die dynamische Konfiguration des Datenbusses mit der $\overline{\text{DSACK0}}$- und $\overline{\text{DSACK1}}$-Leitung. Man kann damit einen 8-, 16- oder 32-Bit-Datenbus definieren.

Ein Buszyklus ist die minimale Einheit eines Speicherzyklus, er dauert drei Taktzyklen. Dieser Buszyklus kann asynchron, d. h. völlig losgelöst vom Takt (CLK) oder synchron, also auf den Takt bezogen, abgeschlossen werden *(Bild 5)*. Je nachdem, welche Anwendung in die Schaltung des Entwicklers hineinpaßt, wählt er eine der Varianten von Bild 5 aus.

Die Verkürzung der Signal-Pfade auf dem Chip-Layout des 68030 ermöglichen nicht nur eine Erhöhung der Standard-Taktfrequenz, sondern reduzieren auch die Zeit, um Adreß- und Datenleitungen zu treiben. Als Resultat konnte der Buscontroller, eines der internen Steuerelemente, verbessert werden. Dieser ist beispielsweise für den externen Buszyklus zuständig. Beim 68030 ist es nun möglich, den Buszugriff bereits in zwei Taktzyklen abzuschließen. Das bedeutet eine Leistungssteigerung um 33% bei gleicher Frequenz und entspricht bei einer Taktfrequenz von 20 MHz einer maximalen Datentransferrate von 40 MByte pro Sekunde! Der 68020 bringt es mit seinem normalen Buszyklus und 16,6 MHz immerhin auf 22 MByte/s. Beim 68030 muß dieser schnelle Buszugriff synchron abgeschlossen werden und die Busbreite ist immer 32 Bit. Ein solcher Buszyklus wird dem Prozessor mit dem neuen Signal $\overline{\text{STERM}}$ (Synchronous TERMination) mitgeteilt *(Bild 6)*.

Der 68030 erlaubt damit eine Mischung der drei möglichen Buszyklen. Er kann auch dynamisch konfiguriert werden, abhängig von $\overline{\text{DSACKx}}$ und $\overline{\text{STERM}}$. Es darf allerdings nur eines dieser Signale gleichzeitig aktiv sein. Für derartig schnelle Applikationen kommen nur Gatter mit geringen Verzögerungszeiten (z. B. TTL-Fast-Serie) und sehr schnelle Speicher (Zugriffszeiten etwa 25 ns) in Frage, beispielsweise das SRAM MCM6290-25 von Motorola [4]. Kommen langsamere Speicher zum Einsatz, so können mit $\overline{\text{STERM}}$ selbstverständlich wie bei den $\overline{\text{DSACKx}}$-Signalen Wartezyklen eingefügt werden. Die Verwendung von $\overline{\text{STERM}}$ bei einem Standard-Buszyklus mit drei Taktzyklen gibt dem Anwender zusätzliche Zeit (etwa einen Halbzyklus) z. B. für Fehlererkennung/Korrektur, Paritäts-Erkennung usw.

Um Daten noch schneller zu übertragen und gleichzeitig neue Speicherkonzepte zu unterstützen, haben sich die Entwickler noch mehr einfallen lassen: den Burst-Fill-

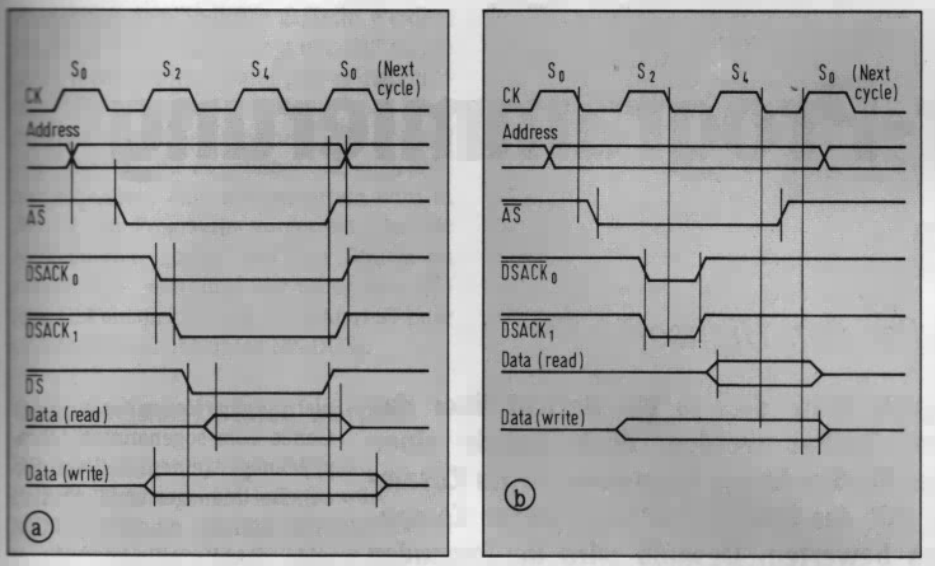


Bild 5. So sehen Buszyklen des 68020 (a) bzw. 68030 (b) aus, wenn sie mit DSACKx abgeschlossen werden

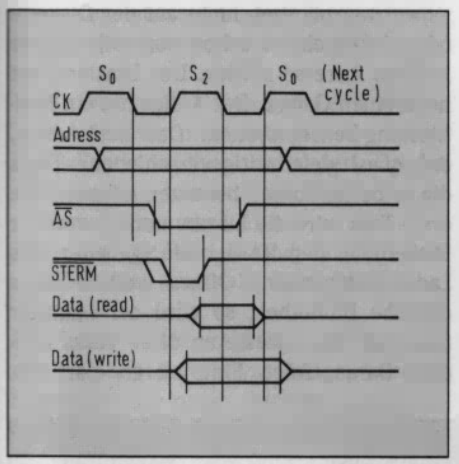


Bild 6. Um einen Taktzyklus schneller ist der 68030 bei einem mit STERM abgeschlossenen Buszyklus

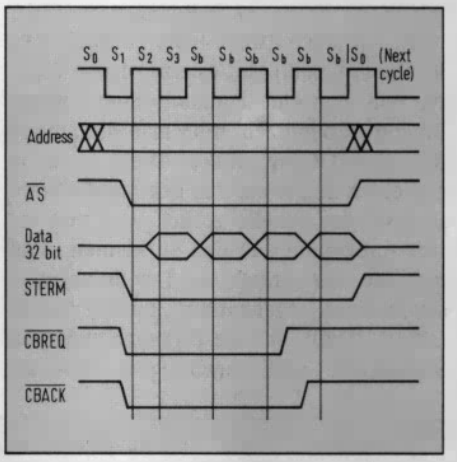


Bild 7. Für den Burst-Fill-Modus ist zusätzliche, extrem schnelle Hardware nötig

Modus. *Bild 7* zeigt den zeitlichen Ablauf. Das Besondere dabei: Vier Langworte (32 Bit) können in minimal fünf Taktzyklen (ohne Wartezyklen) übertragen werden. Standardmäßig braucht man zehn (synchron) oder fünfzehn (asynchron) Taktzyklen. Wenn wir den synchronen Zyklus in Bild 6 gegenüberstellen, erhalten wir eine Datenrate von 64 MByte/s. Dieser Modus wurde möglich durch die Einführung des Datencaches und unterstützender neuer Signale (siehe Bild 3) wie CBREQ (Cache Bus REQest) und CBACK (Cache Bus ACKnowledge):

- CBREQ ist ein Ausgang und wird vom Prozessor aktiviert, wenn er für den Burst-Modus bereit ist.
- CBACK ist ein Eingang und zeigt dem Prozessor an, daß der externe Baustein beim nächsten Taktzyklus ein Langwort liefern wird.

Bestimmte Speicherzugriffe sollen nicht in den (Daten-)Cache übernommen werden, da unter Umständen Daten gelesen werden, die nicht auf dem neuesten Stand sind: So muß in Systemen, bei denen etwa mehrere Prozessoren auf gemeinsame Resourcen (wie gemeinsame Speicherbereiche, Semaphoren oder auch Peripheriebausteine) zugreifen, eine solche Übernahme verhindert werden können. Obwohl die Caches noch nicht beschrieben wurden, betrachten wir in diesem Zusammenhang zwei weitere Signale:

- CIIN (Cache Inhibit IN) ist ein Eingang und zeigt an, daß die Daten des derzeitigen Buszyklusses nicht in den (internen) Cache übernommen werden sollen, ein laufender Burst-Zyklus wird abgebrochen.
- CIOUT (Cache Inhibit OUT) ist ein Ausgang und zeigt an, daß die MMU einen Bereich (Page) angesprochen hat, der nicht in den (internen oder externen) Cache übernommen werden soll. Dieses Signal wird aktiv, wenn das CI-Bit im sogenannten Page-Deskriptor der MMU gesetzt ist und wird zum Sperren eines eventuell extern aufgebauten Caches verwendet.

Je komplexer die Mikroprozessoren werden, desto schwieriger wird die Emulation. Aus diesem Grund bietet der 68030 eigene Signale zur Emulations-Unterstützung, die selbstverständlich allgemein Verwendung finden können. So sind Befehle oder Daten, die im Cache abgearbeitet werden, von außen nicht meßbar. Daher wurde schon beim 68020 die Möglichkeit geschaffen, den Cache abzuschalten. Beim 68030 werden beide Caches gesperrt. Der Inhalt der Caches wird dabei nicht gelöscht, nach der Freigabe steht die alte Information wieder zur Verfügung. Mit MMUDIS (MMU DISable) kann die interne MMU abgeschaltet werden.

STATUS- und REFILL-Signal geben Informationen über den internen Status ab: Das REFILL-Signal zeigt an, wann die interne Pipeline neu geladen wird, hervorgerufen durch eine Programmflußänderung, also z. B. Sprünge oder Verzweigungen. Bei der HALT-Leitung des 68030 fällt auf, daß sie im Gegensatz zum 68020 nicht bidirektional arbeitet. Das bedeutet, daß ein doppelter Busfehler beim 68030 an der STATUS-Leitung abgelesen werden muß. Die Anzahl der Taktzyklen, währen denen das STATUS-Signal aktiv bleibt, gibt nämlich an, ob sich die CPU am Ende eines Befehls befindet (ein Taktzyklus), eine Trace- oder Interrupt-Exception ansteht (zwei Taktzyklen) oder ob eine Adreß-, Busfehler-, Reset-, falscher Interrupt- oder Autovektor-Interrupt-Exception ansteht (drei Taktzyklen). Die beiden Leitungen REFILL und STATUS können daher zur Implementierung einer Echtzeit-Einzelschrittsteuerung (Trace) hergenommen werden [5].

Literatur

[1] *Hilf, W. und Nausch, A.:* M68000-Familie, Band I und II, tewi-Verlag, München.
[2] *Motorola Inc.:* MC68020, 32-Bit-Mikroprozessor User's Manual, Prentice-Hall.
[3] *Hilf, W.:* MC68020, 32-Bit-Prozessor für zukunftssichere Systemkonzepte, Elektronik 14/84, Seite 35 ff.
[4] *Motorola Inc.:* Jones, David Application Note AN... MC68030, 25 MHz Benchmarking Board.
[5] *Motorola Inc.:* MC68030 Enhanced, 32-Bit-Mikroprozessor User's Manual.

Dieser Betrag wird in der nächsten mc mit der Software und den Caches fortgesetzt.

Peter Wollschlaeger

Assembler-Programmierung des Amiga

Der Assembler/Editor DevPac von Highsoft

Ursprünglich sollte dies ja ein Bericht über das Programm DevPac werden, doch gerade einen Assembler für den Amiga kann man nur im Zusammenhang mit der System-Software dieser Computer richtig bewerten. Deshalb wird im folgenden Beitrag auf beide Themen näher eingegangen.

Normalerweise findet man auf jedem Computer eine Routine mit dem Namen CONOUT, die – nach der simplen Regel „Zeichen ablegen, Funktion aufrufen" behandelt – ein Zeichen auf dem Bildschirm ausgibt. Wechselt man den Computer, so muß man sich vielleicht daran gewöhnen, daß das Zeichen nun nicht mehr in D0, sondern auf dem Stack erwartet wird und daß es nicht mehr JSR sondern 'TRAP #1' heißt, aber im Prinzip ändert sich nichts. Es sei denn, Sie wechseln auf den Amiga, da ändert sich alles. CONOUT gibt es nicht, und hat man mühsam herausgefunden, welche Funktion man dafür ersatzweise verwenden kann, so läuft diese auch erst nach nicht unbeträchtlichen Initialisierungen, selbst dann nur unter dem CLI und nicht auf der Workbench und erst recht nicht im Multitasking. Trotzdem ist Assembler-Programmierung auf dem Amiga genau so (wenig) einfach wie die in C, man muß halt nur einiges mehr vorab wissen. Mir persönlich ist Assembler auf dem Amiga sogar sympatischer als C. Nun denken Sie nicht, daß müßte ich als Autor des „Amiga-Assembler-Buches" [1] ja sagen. Mein anderes Amiga-Buch [2] wurde überwiegend in C geschrieben. Bei der Gelegenheit: Als Übersetzer des DevPac-Handbuchs mag ich nicht ganz objektiv erscheinen, doch die zeitliche Abfolge spricht dagegen. Ich habe mich nach dem Test von sechs Assemblern für DevPac entschieden, dann das Assembler-Buch mit dem DevPac geschrieben und erst dann den Auftrag zur Übersetzung des Handbuchs erhalten. Bleibt ein Vorteil für Sie als mc-Leser: Ich bin wesentlich besser mit dem DevPac vertraut, als man es nach den üblichen Software-Tests ist.

Libraries: Schlüssel zum Amiga

Eine Library (oder Bibliothek) ist vereinfacht ausgedrückt eine Sammlung von Unterprogrammen mit einer zugehörigen Sprungtabelle (je Unterprogramm ein Eintrag). Für jeden Zweck (zum Beispiel DOS, Intuition, Grafik) gibt es eine eigene Library. Will man eine Funktion einer solchen Bibliothek benutzen, muß man die Library mit 'OpenLibrary' öffnen. Diese Funktion gibt einen Zeiger auf den Beginn (die Startadresse) der Tabelle zurück. Um nun ein Unterprogramm aufrufen zu können, muß man die Startadresse der Tabelle angeben sowie einen sogenannten „Library Vector Offset" (LVO), der die Differenz zwischen Startadresse und zugehörigem Tabellenplatz angibt. Verwaltet wird das Ganze vom sogenannten Library-Manager (einem Teil von Exec). Der Manager weiß, ob sich eine Library schon im ROM oder RAM befindet. Wenn nicht, versucht er, die Bibliothek von der Diskette zu laden. Klappt das nicht (weil die Bibliothek nicht auf der Diskette oder der Speicher schon voll ist), gibt er Null als Adresse zurück. Der Umstand hat noch einen Grund: Der Amiga hat ein Multitasking-Betriebssystem. Das heißt auch, daß quasi gleichzeitig verschiedene Tasks die selbe Bibliothek benutzen können. Die erste Task wird die Library notfalls von der Diskette in den RAM laden (genauer: das Laden veranlassen). Öffnen weitere Tasks dieselbe Bibliothek, so wird der Manager nur noch die Adresse an diese Tasks melden. Daraus folgt: Eine Library darf erst

```
            incdir  ":devpac/include/"
            include exec/exec_lib.i
            include libraries/dos_lib.i
            include libraries/dos.i

            move.l  #dosname,a1             ;Name der DOS-Lib
            moveq   #0,d0                   ;Version egal
            move.l  _SysBase,a6             ;Basis Exec
            jsr     _LVOOpenLibrary(a6)     ;DOS-Lib oeffnen
            tst.l   d0                      ;Fehler?
            beq     fini                    ;wenn Fehler, Ende
            move.l  d0,a6                   ;Zeiger merken

            jsr     _LVOOutput(a6)          ;Hole Output-Handle

            move.l  d0,d1           ;Handle
            move.l  #msg,d2         ;Puffer-Adr
            move.l  #msglen,d3      ;Max.
            jsr     _LVOWrite(a6)   ;Text ausgeben

            move.l  a6,a1                   ;DOS-Lib-Basis
            move.l  _SysBase,a6             ;Basis Exec
            jsr     _LVOCloseLibrary(a6)    ;Funktion "Schliessen"

fini        rts                             ;Return zum CLI

* Datenbereich:

dosname     dc.b    'dos.library',0
            cnop    0,4
msg         dc.b    'Hallo',10
msglen      equ     *-msg

            end
```

Bild 1. Das muß man schreiben, wenn der Amiga „hallo" ausgeben soll

wieder aus dem Speicher gelöscht werden, wenn die letzte Task gesagt hat, daß sie sie nicht mehr braucht. Dafür gibt es die Funktion 'CloseLibrary'. Jede Task, also jedes Programm, das Sie schreiben, muß deshalb alle Libraries, die es geöffnet hat, auch wieder schließen. Zum Kennenlernen wird in *Bild 1* ein Programm vorgestellt, das die Aufgabe hat, „hallo" auf den Schirm zu schreiben – allerdings nur unter dem CLI (das CLI entspricht TOS beim Atari-ST oder COMMAND.COM unter MS-DOS).

Ohne Include-Dateien wertlos

Die erste Zeile setzt den Suchpfad für das Include-Directory, und damit wären wir beim DevPac. In den Include-Dateien sind tausende von Konstanten, Makros und Offset-Tabellen für die Daten-Strukturen notiert. Ohne diese Include-Dateien ist professionelles Arbeiten mit dem Amiga unmöglich. Drei Punkte sind dem Devpac in diesem Zusammenhang hoch anzurechnen:

1. Es gibt Include-Dateien (die z. B. der Seka-Assembler nicht hat).
2. DevPac kann diese und andere Dateien einbinden (dieses „includieren" beherrscht Seka auch nicht).
3. Die Include-Dateien sind mit dem Amiga-Standard-Assembler kompatibel.

Der letzte Punkt ist sehr wichtig, denn wenn Commodore etwas ändert, erhalten Entwickler die neuen Include-Dateien natürlich im Commodore-Format. Wann (und ob) ein Software-Haus seine Include-Dateien Marke Eigenbau „updated" ist eine andere Frage.
Das Programm will eine Funktion des DOS benutzen und muß deshalb zuerst die DOS-Library öffnen. Für das Öffnen einer Bibliothek benötigt man die Funktion OpenLibrary(), und die ist selbst eine Bibliotheks-Funktion. Für den Zugriff auf eine Bibliothek benötigt man deren Basis-Adresse, und diese erhält man von OpenLibrary() – Sie merken, wir drehen uns im Kreis. Des Rätsels Lösung: Es gibt im Amiga-Betriebssystem eine (und nur eine!) feste Adresse, und das ist die Basis der Exec-Library. Diese Adresse ($4) hat den symbolischen Namen _SysBase. Natürlich haben findige Hacker schon alle möglichen Adressen herausgefunden und greifen fröhlich darauf zu. Sie vergessen dabei nur eines: Das Chaos, das ausbricht, wenn zwei Tasks gleichzeitig auf solche Adressen zugreifen...
Wir haben also mit _SysBase die Basis von Exec und mit _LVOOpenLibrary den Offset in der Sprungtabelle zu dieser Funktion. Hat man die Basisadresse ins Register A6

```
        incdir  ":devpac/include/"
        include exec/exec_lib.i
        include libraries/dos_lib.i
        include libraries/dos.i

* Makros zum File-Handling
* ------------------------

OPEN    macro
        move.l  \1,d1           ;Name
        move.l  \2,d2           ;Mode
        jsr     _LVOOpen(a6)
        endm

RDWR    macro
        move.l  \1,d1           ;Handle
        move.l  \2,d2           ;Puffer-Adr
        move.l  \3,d3           ;Max.
        jsr     _LVO\4(a6)      ;Read oder Write
        endm

CLOSE   macro
        move.l  \1,d1           ;Handle
        jsr     _LVOClose(a6)
        endm

        movem.l  a0/d0,-(sp)            ;Kommandozeile retten

* DOS-Lib oeffnen
* ---------------
        move.l  #dosname,a1             ;Name der DOS-Lib
        moveq   #0,d0                   ;Version egal
        move.l  _SysBase,a6             ;Basis Exec
        jsr     _LVOOpenLibrary(a6)     ;DOS-Lib oeffnen
        tst.l   d0                      ;Fehler?
        beq     fini                    ;wenn Fehler, Ende
        move.l  d0,a6                   ;Zeiger merken


* Kommandozeile holen
* -------------------
        movem.l (sp)+,a0/d0             ;hole Kommandozeile
        move.b  #0,-1(a0,d0.l)          ;terminiere mit 0

* Beide Files oeffnen
* -------------------
        OPEN    a0,#MODE_OLDFILE        ;Quell-File
        move.l  d0,d4                   ;Handle
        tst.l   d0                      ;ging was schief?
        bne     weiter                  ;wenn nicht
        jsr     _LVOOutput(a6)          ;Hole Output-Handle
        RDWR    d0,#msg1,#len1,Write    ;und melde Fehler
        bra     cl_lib                  ;und Ende

weiter  OPEN    #ziel,#MODE_OLDFILE     ;Zielfile (Console)
        move.l  d0,d5                   ;wie vor öffnen
        tst.l   d0
        bne     loop
        jsr     _LVOOutput(a6)
        RDWR    d0,#msg1,#len2,Write
        bra     cl_q


* Hier geht's nun los
* -------------------

loop    RDWR    d4,#buffer,#256,Read    ;Lese 256 Bytes oder weniger
        RDWR    d5,#buffer,d0,Write     ;Ist-Laenge schreiben
        tst.l   d0                      ;EOF?
        bne     loop

        RDWR    d5,#buffer,#1,Read      ;warte auf Taste

cl_z    CLOSE   d5                      ;Close Zielfile
cl_q    CLOSE   d4                      ;und Quellfile

cl_lib  move.l  a6,a1                   ;DOS-Lib-Basis
        move.l  _SysBase,a6             ;Basis Exec
        jsr     _LVOCloseLibrary(a6)    ;Funktion "Schliessen"

fini    rts                             ;Return zum CLI

* Datenbereich:

dosname  dc.b   'dos.library',0
         cnop   0,2
```

Bild 2. Dieses Programm demonstriert den Umgang mit AmigaDOS

geladen (A6 ist übrigens obligatorisch), dann kann man mit

```
jsr offset(a6)
```

eine Funktion aufrufen. Vorher sind natürlich noch die zugehörigen Parameter in Register zu laden. Ich will ja nicht lästern (aber ich tue es trotzdem), doch eines fällt auf:
Sowohl Atari als auch Commodore erklären C zu ihrer Entwicklungssprache. Während man das nun beim Atari als Assembler-Programmierer deutlich merkt und immer brav die Parameter auf den Stack lädt, schreibt man sie beim Amiga in Register. Eine C-Funktion packt aber ihre Parameter auf den Stack, weshalb der vom C-Compiler (nebst Linker) erzeugte Code beim Amiga richtig lustig aussieht, nämlich so: Parameter auf den Stack bringen, gleich wieder vom Stack holen und in Register laden.

Standard-Namen

Nach dem Aufruf von OpenLibrary() steht die Adresse in D0, von wo sie in A6 umgeladen wird. A6 zeigt damit auf die Basis von DOS, mit dem wir nun arbeiten. Normalerweise wird man die Basisadresse einer Bibliothek sofort in eine Variable umladen. Deren Namen beschafft man sich auch aus Include-Dateien. Nun kann man nämlich Makros anwenden, die diese Namen verwenden. Damit wird folgendes Problem gelöst: ein Programm benötigt in der Regel mehrere Bibliotheken und wird bunt gemischt mal eine Funktion aus dieser, mal eine Funktion aus jener Library aufrufen. Vorher ist aber immer A6 mit der richtigen Basis-Adresse zu laden. DevPac unterstützt diese Technik voll: Er stellt die richtigen Namen für die Variablen zur Verfügung, bietet (auch mit den Commodore-Namen) die erforderlichen Makros und behandelt Makros genau wie der Standardasssembler. Alle Assembler, die sich nicht an diese Standards halten (solche Assembler gibt es leider), sind meines Erachtens nur Spielzeug.
Doch nun zum Listing: mit _LVOOutput wird das „Handle“ des Standard-Outputs (hier das CLI-Fenster) ermittelt. Ein Handle ist für Assembler-Programmierer ein Langwort, in C muß man berücksichtigen, daß es ein Zeiger auf eine Datenstruktur ist. Dieses Handle braucht man für die Funktion Write(). Übergibt man ihr noch die Adresse eines Puffers und seine Länge, so schreibt Write() die Zeichenkette aus dem Puffer auf den Schirm, wobei hier eine Standard-Konsole emuliert wird, Zeilen-Umbruch und Scrolling erfolgen hier automatisch.

```
LF        dc.b    10
msg1      dc.b    'Quellfile nicht gefunden',10
len1      equ     *-msg1
          cnop    0,2
msg2      dc.b    'Konnte Console nicht oeffnen',10
len2      equ     *-msg2
          cnop    0,4
quelle    ds.b    40
ziel      dc.b    'CON:0/12/640/200/Console',0
          cnop    0,4
buffer    ds.b    256

          end
```

```
* Menu-Demo

          incdir  ":devpac/include/"

          include intuition/intuition.i
          include intuition/intuition_lib.i
          include exec/exec_lib.i
          include graphics/graphics_lib.i

* Text im Fenster zeichnen
* ------------------------
PRINT     macro
          moveq   \1,d0                   ;X-Position
          moveq   \2,d1                   ;Y
          move.l  windowptr,a1            ;Via Window-Zeiger
          move.l  wd_RPort(a1),a1         ;  Rast-Port-Adresse holen
          move.l  a1,a3                   ;  und retten
          CALLGRAF Move                   ;Funtion Move to X,Y
          move.l  a3,a1                   ;brauche wieder Rastport
          lea     \3,a0                   ;Adresse Text
          moveq   \4,d0                   ;seine Laenge
          CALLGRAF Text                   ;und ausgeben
          endm


MENU      macro
          dc.l    \1                    ;^Nachfolger
          dc.w    \2,\3                 ;x,y
          dc.w    \4,\5                 ;breit, hoch
          dc.w    MENUENABLED           ;Flags
          dc.l    \6                    ;^Titel
          dc.l    \7                    ;^Item-Liste
          dc.w    0,0,0,0               ;System-Use
\6        dc.b    ' \8 ',0              ;Text-String
          cnop    0,2
          endm

ITEM      macro
          dc.l    \1                    ;^next Item
          dc.w    \2,\3                 ;x,y
          dc.w    \4,\5                 ;breit, hoch
          dc.w    ITEMENABLED!ITEMTEXT!HIGHCOMP
          dc.l    0                     ;keine Excludes
          dc.l    \6                    ;^Text
          dc.l    0                     ;Select Fill
          dc.b    0                     ;Cmd-Key
          dc.b    0                     ;Dummy
          dc.l    \7                    ;Sub-Item
          dc.w    0                     ;next Select
\6        dc.b    0,1,RP_JAM1,0         ;Text-Struktur
          dc.w    8,2
          dc.l    0,ttt\6,0
ttt\6     dc.b    ' \8 ',0              ;Text selbst
          cnop    0,2
          endm

          include startup.i             ;auch Start von WB

_main

* Intuition Library oeffnen:
* --------------------------
lea       intname,a1
moveq     #0,d0
CALLEXEC OpenLibrary
tst.l     d0
beq       abbruch
move.l    d0,_IntuitionBase     ;Basis-Zeiger sichern
```

Bild 3. Der Aufwand in Intuition wird nur durch Include-Dateien erträglich

```
* Graphics Library oeffnen
* ------------------------
        lea     grafname,a1
        moveq   #0,d0
        CALLEXEC OpenLibrary
        tst.l   d0
        beq     closeint
        move.l  d0,_GfxBase             ;Basis-Zeiger sichern

* Window oeffnen
* --------------
        lea     windowdef,a0            ;zeige auf Window-Struktur
        CALLINT OpenWindow              ;oeffne Window
        tst.l   d0                      ;ging was schief?
        beq     closegraf               ;wenn ja
        move.l  d0,windowptr            ;Window-Zeiger sichern

* Menu installieren
* -----------------
        or.w    #COMMSEQ,sub12_1+12     ;Cmd-Key setzen
        move.b  #'x',sub12_1+26         ;(wilder Trick)

        move.l  d0,a0                   ;^Window
        lea     menu1,a1                ;^Menu
        CALLINT SetMenuStrip            ;do it

* Auf Event warten, dann auswerten
* --------------------------------
event
        move.l  windowptr,a0            ;zeige auf Window-Struktur
        move.l  wd_UserPort(a0),a0      ;nun auf Message-Port
        move.l  a0,a5                   ;rette Port-Adresse
        move.b  MP_SIGBIT(a0),d1        ;Signal Bit holen
        moveq   #0,d0                   ;Nummer in
        bset    d1,d0                   ;Maske wandeln
        CALLEXEC Wait                   ;Schlaf gut!

        move.l   a5,a0                  ;hole Port-Adresse
        CALLEXEC GetMsg                 ;hole Message
        move.l   d0,a1                  ;muss nach a1
        move.l   im_Class(a1),d4        ;Msg-Typ
        move.w   im_Code(a1),d5         ;Untergruppe
        CALLEXEC ReplyMsg               ;quittiere Msg in a1

        cmpi.l   #CLOSEWINDOW,d4        ;Window Closed?
        beq      closewindow            ;wenn so

        cmpi.l   #MENUPICK,d4           ;Menu angewaehlt?
        bne      event
*------------------------------------ Ende Event-Loop

do_menu
        cmpi     #MENUNULL,d5           ;Item selektiert?
        beq      event                  ;wenn nicht
        cmpi     #$20,d5                ;Exit?
        beq      closewindow
        move     d5,d2                  ;Code
        lea      buffer+8,a0            ;in hex anzeigen
        bsr      hex                    ;Routine aufrufen
        PRINT    #50,#50,buffer,#12
        bra      event

closewindow
        move.l  windowptr,a0            ;wir sind wieder wach
        CALLINT CloseWindow             ;Fenster zu

* Libraries schliessen
* --------------------
closegraf
        move.l  _GfxBase,a1
        CALLEXEC CloseLibrary

closeint
        move.l  _IntuitionBase,a1
        CALLEXEC CloseLibrary

abbruch
        moveq   #0,d0                   ;oder normales Ende
        rts

* Konvertiere d2.w in ASCII-String ab (a0)
* ----------------------------------------
hex
        moveq   #3,d1           ;fuer 4 Nibble
next    rol     #4,d2           ;hole 1 Nibble
        move    d2,d3           ;nach d3 retten
        and.b   #$0f,d3         ;maskiere es
```

AmigaDOS-Programmierung: einfach

Sie haben schon am vorherigen Beispiel gesehen, daß die Programmierung unter AmigaDOS recht einfach ist, jedenfalls nicht schwieriger als das Arbeiten mit MS-DOS oder dem (ziemlich identischen) TOS des Atari-ST. Anzumerken wäre allerdings, daß das CLI dem Kommando-Interpreter von MS-DOS deutlich und dem des Atari-ST haushoch überlegen ist. *Bild 2* zeigt nun schon fast alle Techniken, die man für AmigaDOS-Programme beherrschen muß. Das Programm öffnet ein eigenes Fenster und gibt darin den Text einer Datei aus. Der Dateiname wird mit der Kommandozeile übergeben: Heißt das Programm LIST, so kann eine Datei mit „LIST name“ ausgegeben werden. Der Trick an der Geschichte ist, daß für AmigaDOS auch ein Fenster auf der Konsole eine Datei ist, genau so, wie es Geräte sind, z. B. die Drucker- oder V.24-Schnittstelle. Eine solche Datei wird mit Open() geöffnet, die Funktion gibt ein Handle zurück. Open() selbst benötigt die Adresse einer Zeichenkette (den Namen der Datei) und den Modus. Ist letzterer MODE_OLDFILE, wird eine existierende Datei geöffnet, im Falle von MODE_NEWFILE wird eine neue Datei angelegt bzw. eine vorhandene überschrieben. Die Varianten bietet die Zeichenkette: Hier kann man den vollständigen Pfadnamen einer Disketten-Datei eintragen, einen Geräte-Namen oder eine Bildschirmfenster-Definition. Der Komfort geht soweit, daß die Angabe von „PRT:“ ausreicht, um die Ausgabe an die in Preferences (entspricht dem Kontrollfeld von Macintosh und Atari-ST) eingetragene Schnittstelle zu leiten – und zwar unter Nutzung des gewählten Druckertreibers, der Papierart usw. Sowohl Write() als auch Read() brauchen das Handle, die Adresse des Datenpuffers und die Anzahl der zu schreibenden bzw. maximal einzulesenden Zeichen. Beide Funktionen geben (wie immer in D0) die Anzahl der tatsächlich übertragenen Zeichen zurück. Da sie sich sonst nur durch den Offset beim Aufruf unterscheiden, habe ich beide Funktionen im Makro RDWR zusammengefaßt. Die Kommandozeile (bzw. alles nach dem Programmnamen) befindet sich nach dem Aufruf im RAM. Die Adresse wird in A0, die Länge in D0 übergeben. Da beide Register (wie auch A1 und D1) sogenannte „Scratch“-Register sind (d. h. von jeder Routine überschrieben werden), muß man als allererstes im Programm diese beiden Register retten. Den Rest des Programms kennen Sie schon aus Bild 1. Anzumerken wäre vielleicht noch dies: Nach dem Öff-

```
        add.b   #48,d3          ;in ASCII wandeln
        cmp.b   #58,d3          ;ist es >9 ?
        bcs     out             ;wenn nicht
        addq.b  #7,d3           ;sonst muss es A-F sein
out     move.b  d3,(a0)+        ;1 Zeichen abspeichern
        dbra    d1,next         ;next nibble
        rts

buffer  dc.b    'Code =  xxxx  hex'
        cnop    0,2

W_Title dc.b    'Fenster-Titel',0

windowdef
        dc.w    0,12,640,200
        dc.b    -1,-1
        dc.l    CLOSEWINDOW!MENUPICK
        dc.l    WINDOWCLOSE!ACTIVATE
        dc.l    0,0
        dc.l    W_Title
        dc.l    0,0
        dc.w    0,0,0,0
        dc.w    WBENCHSCREEN

menu1   MENU    menu2,10,0,120,10,MName1,item11,Projekt
menu2   MENU    0,150,0,150,10,MName2,item21,Ich_Dein_Amiga
item11  ITEM    item12,-5,0,120,11,text11,0,Hallo_Welt!
item12  ITEM    0,-5,22,120,11,text12,sub12_1,Exit
sub12_1 ITEM    0,100,10,150,20,text12_1,0,Wirklich?
item21  ITEM    item22,-5,0,120,11,text21,0,Wie_geht's?
item22  ITEM    0,-5,22,120,11,text22,0,Na_Ja

intname         INTNAME         ;Name Intuition Lib (via Makro)
grafname        GRAFNAME        ;Name Graphics Lib

_IntuitionBase  ds.l    1       ;Speicher fuer Zeiger
_GfxBase        ds.l    1
windowptr       ds.l    1
```

nen des Konsol-Fensters folgt ein Test auf Erfolg. Im Fehlerfall wird eine Meldung ausgegeben. Da dann CON nicht offen ist, benutze ich das Output-Handle, das immer existiert (und nie das Opfer von Close() werden darf!). Solche Übungen sind beim Amiga absolute Pflicht, denn man weiß ja nie, ob die anderen Tasks noch genug RAM übrig gelassen haben, um zum Beispiel ein Bildschirmfenster anlegen zu können.
Ein so kurzes Programm erledigt der Assembler DevPac auf der RAM-Disk in Sekundenbruchteilen. Da er wahlweise auch sofort ausführbaren Code erzeugt, ist man drastisch schneller als in C, wo auf den eh schon langsameren Compiler auch noch der Linkerlauf folgen muß.

Keine Resource-Files

Mit *Bild 3* komme ich zu einem „echten" Amiga-Programm mit eigenem Fenster, Pull-Down-Menüs und Mausbedienung und zur ersten Enttäuschung in Sachen Amiga: Der große Frust überkommt einen nämlich, wenn man (wie beim Macintosh oder Atari-ST gewohnt) im Amiga nach Resource-Dateien sucht und keine findet. Tatsächlich sind beim Amiga alle „Resourcen" (wie Fenster, Menüs oder Dialog-Boxen) Datenstrukturen, die im Programm initialisiert werden müssen. In Assembler kann man das Problem zwar mit Makros erheblich mildern, aber viel Tipperei bleibt doch. Da man erst nach dem Programmlauf sieht, was man da gestaltet hat, und das einem natürlich nicht auf Anhieb gefällt, sind mehrere Rekursionen fällig. Hier ist der Assembler mit seinem so deutlich höheren Tempo gegenüber C eindeutig im Vorteil. Inzwischen gibt es die ersten Ansätze von Resource-Editoren, die den Code erzeugen, doch das „Gelbe vom Ei" sind diese Programme noch nicht.

Ohne Makros kaum programmierbar

Bild 4 zeigt als stark gekürzten Auszug aus [2], was hinter den Menüs steckt. Ohne die Makros MENU und ITEM zu Beginn des Listings müßte man für jeden Menü-Titel und jedes Item diese Listen von DC-Direktiven schreiben. Ein echtes Programm mit typisch 4 Menüs und 20 Items würde dann allein dafür 324 „DC"-Zeilen benötigen. Bemerkenswert im Bild 3 ist die Anwendung von Intuition, was dem GEM des ST oder der Toolbox des Mac entspricht. Dazu ist natürlich zuerst die Intuition-Library zu öffnen und dann ein Intuition-Window. Ein solches Intuition-Fenster beinhaltet wesentlich mehr als ein Fenster unter GEM, es ist im Prinzip eine ganze Konsole. Der Grund ist wieder das Multitasking: Es können nämlich beliebig viele Fenster (solange der RAM reicht) gleichzeitig offen sein und in jedem Fenster kann eine andere Task laufen. Das Fenster ist damit auch die Schnittstelle zur Peripherie (Tastatur, Maus usw.). Ein Fenster wird durch den Aufruf von OpenWindow() geöffnet. Parameter ist die Adresse einer NewWindow-Struktur. In dieser Struktur wird unter anderem beschrieben, welche Gadgets (= Bedienelemente) das Fenster erhalten soll (z.B. Close-Gadget, Size-Gadget). Ferner gibt man vor, welche Ereignisse (Events) der Window-Task gemeldet werden sollen. In diesem Beispiel soll das Programm informiert werden, wenn der Anwender das Close-Gadget (CLOSEWINDOW) oder ein Menü anwählt (MENUPICK). Menüs sind immer an ein Fenster gebunden, jedes Fenster kann seine eigenen Menüs haben. Jedes Fenster kann entweder den Workbench-Screen benutzen oder einen vom Programmierer definierten Custom-Screen. Screens sind virtuelle Bildschirme, mit denen unter anderem die Auflösung und die Farben bestimmt werden. Es können gleichzeitig beliebig viele Screens (bis zum Speicherlimit) vorhanden sein.
Die Funktion OpenWindow() liefert einen Zeiger auf eine Fenster-Struktur. Gleichzeitig richtet Intuition zwei sogenannte IDCMPs ein. Das heißt Intuition Direct Communication Message Port. Das Bildschirmfenster empfängt Messages (= Nachrichten) über Events (= Ereignisse) in einen IDCMP namens UserPort. Zu Beginn der Event-Schleife ermittelt das Programm die Port-Adresse: „windowptr" ist der Zeiger auf die Fenster-Struktur. „wd_UserPort" ist der Offset von dieser Adresse auf ein Langwort, in dem die Adresse des Message-Ports steht. Der Message-Port wiederum ist eine Datenstruktur, in der man mit dem Offset „MP_SIGBIT" die sogenannten Signal-Bits findet.
Ein ST-Programmierer ist es gewohnt, in der Event-Schleife zu prüfen, ob eines der erwarteten Ereignisse eingetreten ist. Dieses Abfragen ist zwar auch auf dem Amiga möglich, man sieht es sogar häufig in Listings, es ist aber im Prinzip falsch. So wird nämlich CPU-Zeit damit verbraucht, ständig den Message-Port abzufragen, natürlich tausendmal vergeblich, bis dann endlich das Ereignis eintritt. Diese Zeit fehlt dann den anderen Tasks. Deshalb gibt es (wie in jedem echten Multitasking-System) einen Mechanismus, der (wenn man ihn anwendet) diesen Unsinn stoppt, und das ist die Funktion Wait(). Wait() heißt soviel wie „ich gehe jetzt schlafen, wecke mich, wenn ein Event für mich anliegt". Damit wird die Task auf die Liste der wartenden Tasks versetzt. Worauf sie wartet, teilt sie vorher in den Signal-Bits mit. Leider ist es so, daß im Message-Port eine Bit-Nummer steht, Wait() aber (in D0) eine Maske sehen will,

```
Menüs sind Strukturen die unter anderem einige Zeiger
enthalten, die wiederum auf andere Strukturen zeigen. Für die
Menü-Liste gilt:

Menu 1 --> Menü 2 --> Menü 3 ... Menü n ---> NULL

In jeder Menü-Struktur gibt es auch einen Zeiger auf die Menü-
Titel. Aus diesen wird die Titelleiste aufgebaut. In jedem Menü
gibt es einen Zeiger auf die Item-Liste. Hier gilt dann die
Folge:

Item 1 --> Item 2 --> Item 3 ... Item n ---> NULL

Auch die Items sind Strukturen. Jedes Item hat einen Zeiger auf
die Sub-Item-Liste (oder NULL). für die Sub-Items), gilt wieder:

SubItem 1 --> SubItem 2 --> SubItem 3 .... SubItem n ---> NULL

Innerhalb der Items und Sub-Items gibt es einen Zeiger, der auf
noch eine Struktur zeigt. Diese ist entweder eine Text-Struktur
(mit einem Zeiger auf den Text-String) für den Item-Text oder
eine Image-Struktur, wenn Grafik anstatt Text dargestellt werden
soll. Die Menu-Struktur ist in C so definiert:

struct Menu
{
    struct Menu *NextMenu;      /* Zeiger -> nächstes Menu  */
    SHORT LeftEdge, TopEdge;    /* x,y für linke obere Ecke */
    SHORT Width, Height;        /* Breite und Höhe der Box  */
    USHORT Flags;               /* Flags                    */
    BYTE *MenuName;             /* Zeiger auf Titel-Text    */
    struct MenuItem *FirstItem; /* Zeiger auf Item-Liste    */
    /* Nur für Intuition intern: */
   SHORT JazzX, JazzY, BeatX, BeatY;
};

Für Item und Sub-Items gilt gleichermaßen:

struct MenuItem
{
    struct MenuItem *NextItem;  /* Zeiger auf nächstes Item   */
    SHORT LeftEdge, TopEdge;    /* x,y für linke obere Ecke   */
    SHORT Width, Height;        /* Breite und Höhe der Box    */
    USHORT Flags;               /* Flags                      */
    LONG MutualExclude;         /* Gegenseitiger Auschluß     */
    APTR ItemFill;              /* Zeiger auf Text oder Image */
    APTR SelectFill;            /* Text/Image bei Auswahl     */
    BYTE Command;               /* Command-Key                */
    struct MenuItem *SubItem;   /* Zeiger auf Sub-Item-Liste  */
    USHORT NextSelect;          /* wenn         Mehrfachwahl  */
};
```

Bild 4. Beim Amiga sind Strukturen und Listen das A und O

in der dieses Bit gesetzt ist (so wird z. B. aus Bit 3 binär 1000), daher die Umwandlung mit „bset". Hat ein Programm mehrere Fenster, so müssen deren Masken durch ein Oder verknüpft werden.

Nach „CALLEXEC Wait" schläft also unsere Task. Nur wenn sie geweckt wird, geht es mit den Folgezeilen weiter. Diese lesen die Nachricht, speichern sie und quittieren mit „ReplyMsg()". Danach wird – wie beim ST oder Mac auch – ausgewertet, welches Ereignis vorliegt und entsprechend reagiert. Hier wird im Falle eines Menü-Events nur das „Menü-Wort" als Hexzahl angezeigt. Sinn der Übung ist es, aufzuzeigen, daß in den Bits 0...4 das Menü, in 5...10 das Item und in 11...15 das Sub-Item codiert sind. Ziemlich zu Anfang wird „startup.i" eingezogen. Ohne diesen Modul kann ein Programm nicht von der Workbench aus gestartet werden. In [1] habe ich einen „startup" genau beschrieben, hier nur soviel: Die Routine muß feststellen, ob das Programm von der Workbench oder vom CLI aus aufgerufen wurde. Im Falle „Workbench" meldet es sich als neue Task an und wartet auf die Starterlaubnis der Workbench. Zum Schluß muß es sich wieder abmelden.

Editor/Assembler mit Komfort

Um wieder auf DevPac zurückzukommen: Ich hoffe, Sie nun davon überzeugt zu haben, daß der Amiga ohne Include-Dateien und mächtige Makros nicht professionell zu programmieren ist. Genau das bietet DevPac. Hinzu kommt ein zweiter wesentlicher Punkt: Aufgrund des Zwangs zur interaktiven Programmierung der Benutzeroberfäche sollten immer nur Sekunden zwischen einer Änderung im Editor und dem Testlauf des ausführbaren Programmes liegen. Hier beweist DevPac als integrierter Editor/Assembler seine Stärke: Im Editor reicht ein Tastendruck, um das Programm zu übersetzen und wahlweise gleich zu linken. Das dauert für das Programm von Bild 3 auf einer RAM-Disk ca. zwei Sekunden. Beachten Sie, daß dabei aufgrund der Include-Dateien über 2000 Zeilen assembliert werden. Die Tempo-Bedingung erfüllt DEvPac also allemal.

Daß sich der Assembler nicht nur an den Amiga- sondern auch an den Motorola-Standard hält, muß leider erwähnt werden: Es gibt nämlich Assembler für den Amiga, die nicht einmal letzteres können.

Der Editor ist sehr komfortabel. Man kann ihn kurz aber treffend als „Wordstar mit Mausbedienung" beschreiben. Natürlich gibt es einige Zusätze zum Assembler-Teil, so können etwa im Editor alle Zeilen schrittweise oder direkt angewählt werden, in denen der Assembler Fehler fand. Die Fehlermeldungen werden dabei eingeblendet.

Komfortabler Debugger

Der Debuger heißt MonAm, ich übersetze das immer mit 'mon ami', weil mir „mein Freund" schon aus mancher Klemme geholfen hat. MonAm ist ein symbolischer Debugger, er kommt aber natürlich auch ohne Symboltabelle aus. MonAm bietet alle Monitor-Funktionen, kann disassemblieren, tracen, Breakpoints behandeln, kurz, alles, was gute Programme dieser Art so bieten, ist vorhanden. Besonders angenehm ist dabei, daß man mit MonAm praktisch nie abstürzen kann: Alle 'Gurus' werden abgefangen, sprich die Exeception-Vektoren werden auf eigene Behandlungsroutinen 'verbogen'. Konzeptbedingt kann MonAm ein Programm (Task) nur laufen (tracen) lassen, wenn die Task 'schläft'. Der Versuch der Zuwiderhandlung läßt die Meldung 'Task must be suspended' erscheinen. Im Trace- oder Step-Modus werden in verschiedenen Fenstern die Registerwerte, ein Speicherauszug in hex und ASCII (sinnvollerweise für den Datenbereich) und ein Stück des Quelltextes mit einem Pfeil auf die aktuelle Zeile dargestellt. Alles in allem heißt das, man kann zu jedem Befehl direkt ablesen, wie er auf Variable und Register wirkt.

Literatur

[1] *Peter Wollschlaeger:* Amiga Assembler-Buch, Verlag Markt & Technik, München, 1987

[2] *Peter Wollschlaeger:* Amiga-Programmierpraxis Intuition, Verlag Markt & Technik

[3] *Frank Kremser, Lörg Koch:* Amiga Programmierhandbuch, Verlag Markt & Technik

[4] *Commodore Amiga Inc.:* AmigaDOS-Handbuch, Verlag Markt & Technik

[5] *HiSoft:* Deutsches Handbuch zum DevPac, Verlag Markt & Technik

Günther Sternberg

386-PC höchster Leistung

Der Compaq 386/20 setzt neue Maßstäbe. Erstmals wird in einem PC Intels Cache-Controller 82385 eingesetzt. Eine Technik, die früher nur Minicomputern und Workstations vorbehalten war, erhöht den Systemdurchsatz. mc vergleicht den Compaq 386/20 mit anderen 80386-Maschinen.

Als Compaq 1982 gegründet wurde, meinten viele, daß lediglich ein weiterer Nachbauer von IBM-kompatiblen PCs sein Glück in dem schier unerschöpflichen Terrain der PCs versuchen wollte. Bald setzte sich Compaq von der Masse der Clone-Hersteller ab und verwirklicht seitdem eigene Ideen in der Entwicklung leistungsfähiger PCs.

Die Redaktion war daher gespannt, das neue PC-Flaggschiff, den Compaq 386/20, zu testen. Man bekommt schließlich nicht alle Tage einen PC mit einer 20-MHz-80386-CPU, 5 MByte Hauptspeicher, 32 KByte Cache, einer 60-MByte-Festplatte mit einer mittleren Zugriffszeit von unter 30 ms, einer VGA-Bildschirmkarte und einem VGA-Monitor. Der eingebaute numerische Coprozessor 80387/20 von Intel bildet das I-Tüpfelchen der Hardware. Außer MS-DOS 3.3 lieferte Compaq mit dem Testgerät noch das multitaskingfähige Windows 386 von Microsoft mit.

Bild 1. Der Compaq 386/20 mit 60 MByte-Festplatte, 40 MByte-Streamer und 14-Zoll-Farbmonitor

Einen ersten Eindruck über die Leistungsfähigkeit von 386er-Maschinen in punkto Geschwindigkeit erhält man bereits beim DIR-Befehl. Konnte man bei einem längeren Inhaltsverzeichnis auf einem PC mit 4.77 MHz Taktfrequenz noch gemütlich mitlesen, wurde der gleiche Vorgang auf einem schnellen AT bereits zu einem Reaktionstest bei der Benutzung der CTRL-S-Taste.

Beim Compaq 386/20 muß man beim DIR-Befehl die /P-Option verwenden – oder das Directory rauscht in Sekundenbruchteilen am Bildschirm vorbei.

Der Compaq 386/20 läßt die Muskeln spielen

Der Eindruck von der Reaktion des Systems auf den DIR-Befehl ist jedoch völlig subjektiv. Um objektive Aussagen zu machen, hat mc die Testprogramme PRIM1 und PRIM2 [1] entwickelt. Sie ermitteln die Arbeitsgeschwindigkeit eines Rechners mit und ohne Bildschirmausgabe. Meist bildet die Bildschirmausgabe, vor allem bei CGA- und EGA-Karten, den eigentlichen Flaschenhals eines PCs. Der 8-Bit-E/A-Bus wirkt im Gegensatz zum 32-Bit-Systembus wie ein Nadelöhr. In *Bild 2* sind die Ausführungszeiten von PRIM1 und PRIM2 von verschiedenen PCs dargestellt. Interessant ist der Vergleich mit den 386er-Rechnern anderer Hersteller. Obwohl die Taktfrequenz der anderen 386-Computer nur 4 MHz langsamer ist, ist der Compaq 386/20 fast doppelt so schnell. Der Geschwindigkeitsvorteil dürfte größtenteils an dem Cache liegen. Auch die Laufzeiten der Programme „Sieb des Eratosthenes" und „Fibonacci" [2] in *Bild 3* untermauern die geschwindigkeitsmäßige Überlegenheit des Compaq 386/20. Anzumerken ist noch, das auf allen Maschinen das gleiche Testprogramm lief, also bei 386er-Rechnern kein spezieller 32-Bit-Code verwendet wurde.

Die Leistungsfähigkeit des Compaq 386/20 machte sich nicht nur bei den Benchmarks bemerkbar, sondern trat auch während der 10tägigen Testdauer bei der täglichen Arbeit angenehm in Erscheinung.

Der Cache-Controller 82385

Ab einem gewissen Zahlenwert verkümmert die Angabe der Taktfrequenz einer CPU zu einer aussagelosen, effekthaschenden Zahl im Datenblatt. Entscheidend für die Geschwindigkeit eines Systems ist nicht (nur) die Taktfrequenz der CPU, sondern die Leistungsfähigkeit des Systembusses, d. h. der Busdurchsatz. Speicher mit mehr als 1 MByte Kapazität sind aus Platz- und Kostengründen praktisch nur mit dynamischen Speicherbausteinen zu realisieren. Zwischen der von modernen Prozessoren geforderten Speicherzugriffszeit und der mit dynamischen Speichern möglichen Zugriffszeit klafft eine deutliche Lücke. Selbst Speicherbausteine mit 100 ns Zugriffszeit sind für einen mit 16 MHz oder 20 MHz getakteten 80386-Prozessor bereits viel zu langsam. Was nutzt einem die hohe Prozessorgeschwindigkeit, wenn die CPU bei jedem Speicherzugriff zwangsweise mehrere Wartezyklen einlegen muß? Zwar gelingt es durch Verwendung einiger spezieller Formen der Speicherorganisation, z. B. Interleaved Memory [3], die Zugriffszeit zu vermindern, doch läßt sich damit das Problem nicht beseitigen. Nach dem heutigen Stand der Technik ist ein Cache-Speicher wohl die beste Lösung.

Worin liegt die Aufgabe und der Vorteil eines Cache-Speichers? Führt die CPU einen Speicherzugriff durch, möchte sie möglichst schnell die gewünschte Information bekommen bzw. abspeichern. Zwischen den Speicherzugriffen benötigt die CPU einige Zeit zur Decodierung und Abarbeitung des Befehls, ohne daß auf den Spei-

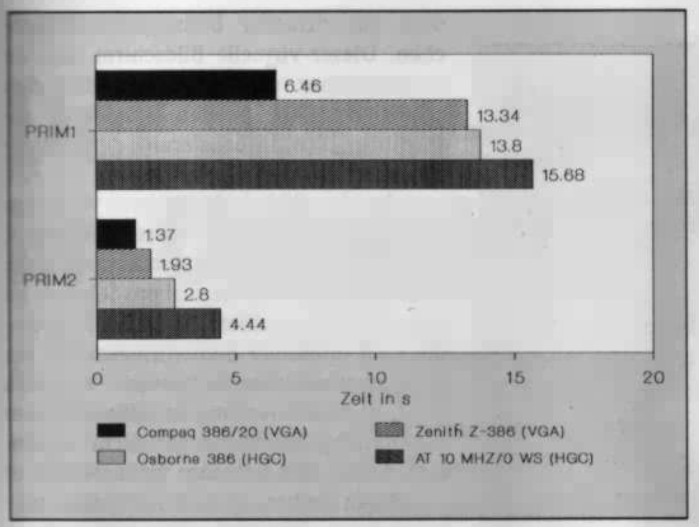


Bild 2. Der Compaq 386/20 hebt sich deutlich von anderen 80386-Maschinen ab

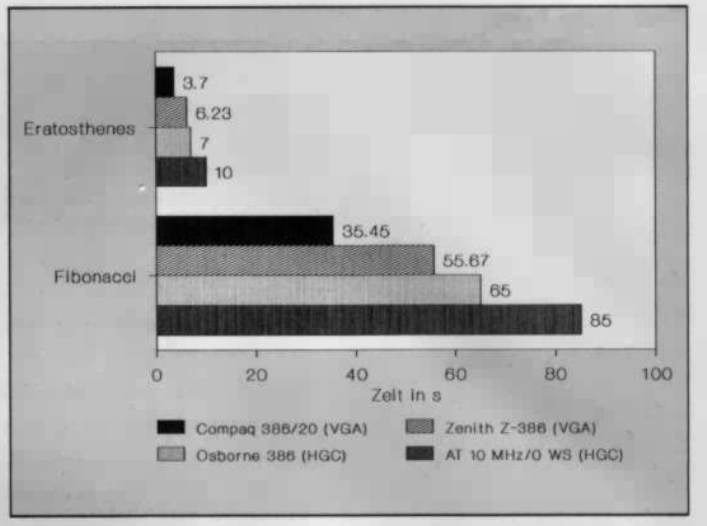


Bild 3. Die Ausführungszeiten der mc-Testprogramme „Sieb des Eratosthenes" und „Fibonacci"

cher, abgesehen von Refresh-Zyklen, zugegriffen wird. Der Cache-Controller, das Herzstück des „Cache-Speichers", nutzt die Zeit, um die Informationen aus der voraussichtlich nächsten vom Prozessor adressierten Speicherstelle des dynamischen Speichers in den Cache-Speicher zu transferieren. Führt der Prozessor dann einen Lesezugriff auf eine Speicherstelle aus, deren Inhalt sich bereits im Cache-Speicher befindet, kann der Cache-Controller dem Prozessor die gewünschte Information so schnell zur Verfügung stellen, daß dieser keine Wartezyklen benötigt. Bei einem Schreibzugriff des Prozessors übernimmt zunächst der Cache-Controller Adresse und Datenwort und führt dann selbst den langsamen Schreibzugriff im Hauptspeicher durch. In der Theorie scheint das Ganze perfekt zu sein.

Woher „weiß" der Cache-Controller auf welche Speicherstelle der Prozessor als nächstes zugreifen wird? Jeder Zugriff auf eine Speicherstelle, deren Inhalt sich nicht im Cache-Speicher befindet, vermindert den Nutzen des Cache-Konzepts und erhöht die Zugriffszeit. Leider ist die Entwicklung von „Cache-Strategien" zum Erreichen möglichst hoher „Trefferquoten" eine Wissenschaft für sich und kann hier nicht näher erläutert werden.

Als Befürworter einer strukturierten und modularen Programmierweise kann ich mir an dieser Stelle eine Anmerkung nicht verkneifen. Jeder für die Lösung einer Aufgabe unnötige Sprungbefehl verringert die Effizienz des Cache-Speichers und sollte daher vermieden werden.

Im Compaq 386/20 wurde der 32-Bit-Cache-Controller 82385 von Intel [4], zusammen mit einem 32 KByte großen statischen Cache-Speicher (35 ns Zugriffszeit), eingesetzt. Ein 80386-System ohne Cache ist jedoch keineswegs schlecht oder minderwertig. Ein Cache ist zwar das Mittel, um das Letzte aus dem System „herauszukitzeln", jedoch erhöht er den Preis eines Systems gewaltig.

Die technischen Daten überzeugen

Alle wichtigen und interessanten technischen Daten sind in der *Tabelle* zusammengefaßt. Erwähnenswert ist die Tatsache, daß neben dem nummerischen Coprozessor 80387 von Intel auch der noch leistungsfähigere Coprozessor 1167 von Weitek alternativ einsteckbar ist. Der 14-Zoll-Video-Grafik-Farbmonitor mit einer Grafikauflösung von 640 x 480 Punkten und einer Textauflösung von 720 x 400 Punkten (bei 256 Farben) besticht im Zusammenspiel mit dem VGA-kompatiblen „Compaq Video Display Controller Board" mit seinen hervorragenden Bildeigenschaften. Auch im VGA-Grafikmodus mit 640 x 480 Punkten zeigt sich ein verzerrungsfreies und klares Bild. Endlich liefert ein PC-Hersteller eine VGA-kompatible Karte mit einer 16-Bit-Schnittstelle aus.

Das Gehäuse des Compaq 386/20 bietet Platz für vier von außen zugängliche Slim-Line-Laufwerke.

Weicher Tastaturanschlag

Bei der Hardware fielen vor allem zwei Punkte unangenehm auf. Die Geräuschentwicklung von Lüfter und Festplatte liegt über dem Lärmpegel von AT-Rechnern und ist auf die Dauer etwas störend. Ein weiteres Manko ist die Tastatur. Die Tastatur entspricht zwar weitgehend der MF-II-Tastatur von IBM, besitzt jedoch einen relativen weichen Anschlag. Die Plazierung des Tastaturkabelsteckers im 90-Grad-Winkel an der Längsseite der Tastatur ist etwas unglücklich. Wirklich störend sind die seitlich nach innen klappenden Stützen zum Aufstellen der Tastatur. Beim Verschieben der Tastatur nach rechts oder links klappen die Stützen ständig ein. Wer seine Tastatur am Schreibtisch öfters hin- und herschieben muß, verzichtet ziemlich schnell auf die Stützen. Das deutsche Benutzerhandbuch enthält zwar alle nötigen Informationen zum System, ist aber nicht mit tiefschürfenden Details versehen. Das mit dem Testgerät mitgelieferte Windows 386 bestand außer den Disketten, nur aus drei dünnen englischen Heftchen, die außer der Installationsanweisung nur spärliche Nebeninformationen enthielten.

Inklusive Windows 386

Über den Wert und Nutzen von grafischen Benutzeroberflächen kann man sich streiten. Für jemanden, der zum ersten Mal oder nur gelegentlich mit dem Computer arbeitet, ist eine grafische Oberfläche sicherlich eine Hilfe. Ich persönlich kann mich mit Windows nicht so recht anfreunden. Bei Windows 386 war ich aber trotzdem neugierig, denn diese Version von Windows ist speziell auf die Möglichkeiten von 386er-Rechnern zugeschnitten: sie ist multitaskingfähig. Damit sollte es beispielsweise möglich sein, im Vordergrund einen Brief zu schreiben, während im Hintergrund ein Autorouter das Layout einer Platine berechnet. Zudem sollte man beliebig zwischen den einzelnen Prozessen (Tasks) hin- und herwechseln können. Wer die große Leistungsfähigkeit des Compaq 386/20 nicht für eine spezielle Anwendung, z. B. im CAD-Bereich oder für intensive numerische Berechnungen benötigt, der kann nur mit einem Multitasking-Betriebssystem die

Technische Daten

Typ	Compaq 386/20
Hersteller	Compaq
CPU	80386
Taktfrequenz	20 MHz
Cache-Speicher	32 KByte mit 35 ns Zugriffszeit
Coprozessor	80387 Coprozessor von Intel mit 20 MHz Taktfrequenz und/oder 1167 Coprozessor von Weitek
Speicher	
– Standard-RAM-Kapazität	1 MByte dynamisches RAM
– maximale Speicherkapazität	16 MByte
– Besonderheiten	16 MByte RAM ohne Verlust eines Steckplatzes
Disketten-Laufwerk	
– Standard	1,2 MByte, 5,25 Zoll
– Optionen	360 KByte, 5,25 Zoll 1,44 MByte, 3,5 Zoll
Festplatten-Laufwerk	
– Modell 60	60 MByte, halbe Höhe, mittlere Zugriffszeit < 30 ms
– Modell 130	130 MByte, volle Höhe, mittlere Zugriffszeit < 20 ms
– Modell 300	300 MByte, volle Höhe, mittlere Zugriffszeit < 20 ms
optionale Streamer	
– Kapazität	40 oder 135 MByte
– Transferrate	1 oder 5 MByte pro Minute
Steckplätze	
– Modell 60	ein 32-Bit-Steckplatz für Arbeitsspeicher zwei 8-Bit-Steckplätze (1 x kurz und 1 × lang) vier 8/16-Bit-Steckplätze mit voller Länge
– Modell 130/300	ein 32-Bit-Steckplatz für Arbeitsspeicher zwei 8-Bit-Steckplätze (1 × kurz und 1 × lang) drei 8/16-Bit-Steckplätze mit voller Länge
Schnittstellen	eine parallele Schnittstelle (25 Pins) eine serielle Schnittstelle (9 Pins)
Bildschirmkarte	
– Bezeichnung	Compaq Video Display Controller Board
– Bildschirmmodi	CGA-, EGA- und VGA-kompatibel
Farbmonitor	
– Bezeichnung	Compaq Video-Grafik-Farbmonitor
– Größe	14 Zoll (Bildschirmdiagonale)
– Auflösung	Grafik: 640 × 480 Punkte Text: 720 × 400 Punkte
– Farben	256 Farben gleichzeitig
Betriebssystem	MS-DOS Version 3.3
sonstige Software	Microsoft Windows 386
Netzgerät-Leistung	192 Watt Dauerleistung
Gehäuseabmessungen	50,3 × 41,9 × 16,2 cm
Garantie	12 Monate
Sonstiges	Echtzeituhr Sicherheitsschloß

Leistungsfähigkeit der Maschine wirklich sinnvoll nutzen.

Die Installation von Windows auf der Festplatte war dank des komfortablen, menügesteuerten Setup-Programms einfach. Nach einer Viertelstunde zur Wiederauffrischung meiner alten Windows-Kenntnisse war ich fit genug, um eine Antwort auf meine drängendste Frage zu erhalten.

Wieviel Kapazität des Rechners „schluckt" bereits Windows allein? Bleibt mit einer Task im Hintergrund zum Schluß für die Aufgabe im Vordergrund eine geringere Leistung als auf einem AT übrig? Zur Klärung der Frage wurde eine länger laufende Version des Programms „Sieb des Eratosthenes" dreimal als Hintergrundtask in je einem Fenster gestartet. Da alle drei Programme gleichberechtigt nebeneinander liefen, sollte die Ausführung eines Programms mindestens dreimal soviel Zeit beanspruchen, als ob das Programm als einziges Programm im Rechner auf der DOS-Ebene liefe. Seltsamerweise benötigten die Programme jedoch weniger als die dreifache Rechenzeit. Die Lösung des Rätsels liegt im „Flaschenhals" Bildschirmausgabe. Das „Eratosthenes"-Programm gibt bei jedem Durchlauf am Bildschirm die neu gefundene Zahl aus und greift über den entsprechenden Interrupt, sehr häufig auf die Bildschirmkarte zu. Unter Windows 386 wird über den gleichen Interrupt nicht direkt der Bildspeicher der Grafikkarte, sondern ein virtueller Bildschirm angesprochen. Dieser virtuelle Bildschirm liegt im Hauptspeicher und ist damit über den schnellen 32-Bit breiten Systembus ansprechbar. Die Aktualisierung der Bildfenster aller Tasks, also das Kopieren des entsprechenden Bildauschnitts vom virtuellen Bildschirm in das reale Fenster übernimmt Windows. Da es vollkommen ausreicht, die Bildfenster drei- bis viermal pro Sekunde zu aktualisieren, resultiert daraus bei Programmen mit intensiver Bildschirmausgabe, ein nicht unerheblicher Zeitgewinn. Wie man sieht, ist die Beurteilung der Effizienz einer Multitasking-Oberfläche gar nicht so einfach. Soviel läßt sich aber wohl sagen, der Verlust an Rechenkapazität zur Taskverwaltung durch Windows 386 hält sich in durchaus akzeptablen Grenzen.

Einschränkungen unter Windows 386

Bis zu diesem Zeitpunkt war der Eindruck von Windows 386 durchaus positiv. Als ich jedoch ein CAD-Programm startete, war es vorbei mit der Freude. Beim Programmstart öffnete sich ein Fenster, daß einem sinngemäß folgendes mitteilte: Durch die Art des Zugriffs auf die Bildschirmkarte kann dieses Programm weder in einem Fenster, noch gleichzeitig zusammen mit anderen Programmen ablaufen. Was ist geschehen? Das CAD-Programm greift – wie fast alle Grafikprogramme – direkt auf den Bildspeicher und die Register der Grafikkarte zu. Ein direkter Zugriff auf die Bildschirmkarte scheint aber unter Windows nicht zulässig zu sein. Genauere Informationen über die Hintergründe der Inkompatibilität des Programms mit Windows gingen aus der mitgelieferten Dokumentation zu Windows leider nicht hervor.

Der Porsche unter den PCs

Der Compaq 386/20 ist wegen seiner Leistungsfähigkeit und seiner technischen Eigenschaften ein Spitzenprodukt. Anwender, die wirklich ein Maximum an Leistung benötigen, dürften mit diesem Gerät bestens bedient sein. Genutzt werden die Eigenschaften des Systems jedoch erst von einem Multitasking-Betriebssystem. Das mitgelieferte Windows 386 ist nur als Übergangslösung anzusehen.

Literatur

[1] *Strauß, Dieter:* Primzahlen-Sucher, mc 1987, Ausgabe 10, Seite 74.

[2] High-Tech Horsepower, Byte Juli 1987, S. 101.

[3] *Strauß, Dieter:* 80386-Power von Osborne, mc 1988, Ausgabe 2, Seite 117.

[4] Intel, Architectural Overview, 82385 High-Performance 32-Bit-Cache-Controller, Publication Order Number 296198-001.

Rolf Schäfer

Der Know-how-Computer im Unterricht

Der Know-how-Computer ist ein Papierformular, das das Schema eines Computers darstellt. Als didaktisches Experiment wurde er von Wolfgang Back und Ulrich Rohde im Rahmen der WDR-Fernsehsendung Computerclub entwickelt, und er hat sich durchgesetzt.

Als Mitte des Jahres 1983 die Serie Papier-Computer in mc gestartet wurde, war unsere Schule für den Informatik-Unterricht denkbar schlecht gerüstet. Lediglich ein Satz programmierbarer Basic-Taschenrechner stand zur Verfügung. Um etwas Abwechslung in den Unterricht zu bringen, startete ich versuchsweise eine Unterrichtseinheit „Know-how-Computer". Dabei hielt ich mich im wesentlichen an das Konzept, das in der mc-Serie vorgegeben war. Mit gutem Erfolg. Die Schüler waren sehr motiviert und lagen oft im Wettstreit, wer ein Problem mit den wenigsten Programmschritten erledigen konnte. Allerdings sollte man auch hier nicht verschweigen, daß nach einer Zeit der Euphorie wieder Ernüchterung eintrat, weil einerseits das Austesten von Programmen allein auf Papier sehr aufwendig ist und andererseits das Debuggen jeweils einen enormen Radieraufwand darstellt. Dennoch startete ich auch in den folgenden Jahren immer wieder eine kurze Unterrichtseinheit „Papiercomputer", und an den insgesamt positiven Erfahrungen änderte sich nichts. *Bild 1* zeigt den Papiercomputer verkleinert.

Im Jahr 1986 wurde dann unsere Schule endlich mit fünf PC 10 von Commodore ausgestattet. Doch wie kann man Schüler mit dem Umgang des für sie völlig ungewohnten Gerätes vertraut machen? Die Richtlinien für den Informatikunterricht sehen für diese Eingewöhnungsphase die Arbeit mit Textverarbeitungs- oder Tabellenkalkulationsprogrammen vor. Das hat aber den Nachteil, daß für die Schüler das Gerät ebenso unbekannt ist wie das Programm. Ich entschied mich deshalb für einen anderen Weg. Da die Thematik „Papiercomputer" den Schülern aus einer unmittelbar vorher stattgefundenen Unterrichtseinheit bekannt war, entschloß ich mich, ein Programm „Know-how-Computer" zu schreiben und zur Einführung zu nutzen. Vorteil bei dieser Vorgehensweise war, daß die Schüler im Bestreben, sehr rasch einen Programmiererfolg zu erzielen, sehr motiviert an die Sache herangingen.

Anleitung für das Know-how-Programm

```
I )   Auf das Laufwerk bzw. das Verzeichnis umschalten,
      in dem Know-How Programme stehen bzw. in das Programme
      geladen werden sollen.

II)   Den File KNOWHOW.COM laden.

III)  Erläuterung der Menue_punkte:

1 Prg Editor: Erlaubt sind die Befehle:
      +r , -r , 0r , jn , stop  wie in Mc beschrieben.

      Zusätzlich:
      cn   Call Unterprogramm
      *    No Operation

      Nach der Eingabe des Befehls kann mit der Return- oder
      mit den Pfeiltasten quittiert werden.
      Mit Control-Links /Rechtspfeil gelangt man in die benach-
      barten Programmspeicher-Spalten.
      Mit Insert/Delete Taste werden Befehle gelöscht bzw.
      eingefügt.
      Mit der Eingabe von "u" wird ein Unterprogramm einge-
      leitet. Beim Programmlauf werden die Rückkehradressen
      automatisch eingefügt und (invers) angezeigt.
      Bei fehlerhafter Eingabe erfolgt eine Fehlermeldung.
      Mit der End-Taste gelangt man zurück ins Hauptmenue.

2 Reg Editor: Die Eingabe der Registerwerte erfolgt entsprechend
      der Eingabe der Befehle.

3 Run Quick : Schneller Programmlauf. Kann (z.B. bei  Endlos-
      schleifen)  jederzeit mit jeder Taste unterbrochen werden.

4 Run Slow  : Programmlauf im Einzelschritt. Jeder Befehl wird
      einzeln abgearbeitet und angezeigt. Die Auslösung eines
      neben Befehls erfolgt mit F1 der vorzeitige Programmab-
      bruch mit F10.

5 Exec Plan : Der Programmlauf wird Schritt für Schritt auf dem
      Drucker dokumentiert.Die Rückkehr aus einem Unterprogramm
      wird mit RET angezeigt. Vor der Programmausführung wird
      eine Titeleingabe für die Dokumentation angefordert. Sie
      dient nur der Übersicht und hat für den Programmlauf keine
      Bedeutung.

6 Clear All : Nach Rückfrage werden Programmspeicher und Register
      gelöscht.

7 Directory : Gemäß Voreinstellung (siehe hierzu Punkt I) wird
      das Inhaltsverzeichnis (ohne Unterverzeichnisse) gelistet.

8 Save Prg  : Programme werden gesichert

9 Load Prg  :                bzw. zurückgeladen.

0 Exit      : Nach Rückfrage Programmabbruch.

IV) Fehlermeldungen erfolgen

    a) bei der Eingabe unzulässiger Befehle.

    b) während des Programmablaufs

       .. wenn der Programmcounter größer als 75 wird.

       .. wenn ein Register kleiner Null wird.

       .. wenn ein Register größer Tausend wird.

       .. wenn das Programm auf eine leere Speicher-
          stelle trifft.
```

Das Programm „Know-how-Computer"

Als ich das Projekt Papiercomputer in Angriff nahm, ahnte ich noch nicht, welcher Programmieraufwand dahintersteckte. Und sehr oft, als ich glaubte, endlich fertig zu sein, wurde ich von Schülern oder Kollegen angehalten, dies oder jenes Feature zusätzlich vorzusehen. So kamen schließlich 1300 Turbo-Pascal-Zeilen zusammen, entwickelt auf einem IBM-kompatiblen Computer (MS-DOS). Der Bildschirm ist in drei Bereiche aufgeteilt: Die Programmspeicherzone, die Registerzone und die Menüleiste mit 10 Auswahlmöglichkeiten. *Bild 2* zeigt den Bildschirmaufbau und ein längeres Programm. Es sind folgende Befehle erlaubt:

+r
addiere 1 zum Register r
–r
subtrahiere 1 von Register r
0r
wenn Register r = 0, dann erhöhe PC um 2, sonst um 1
jn
setze Pc auf n

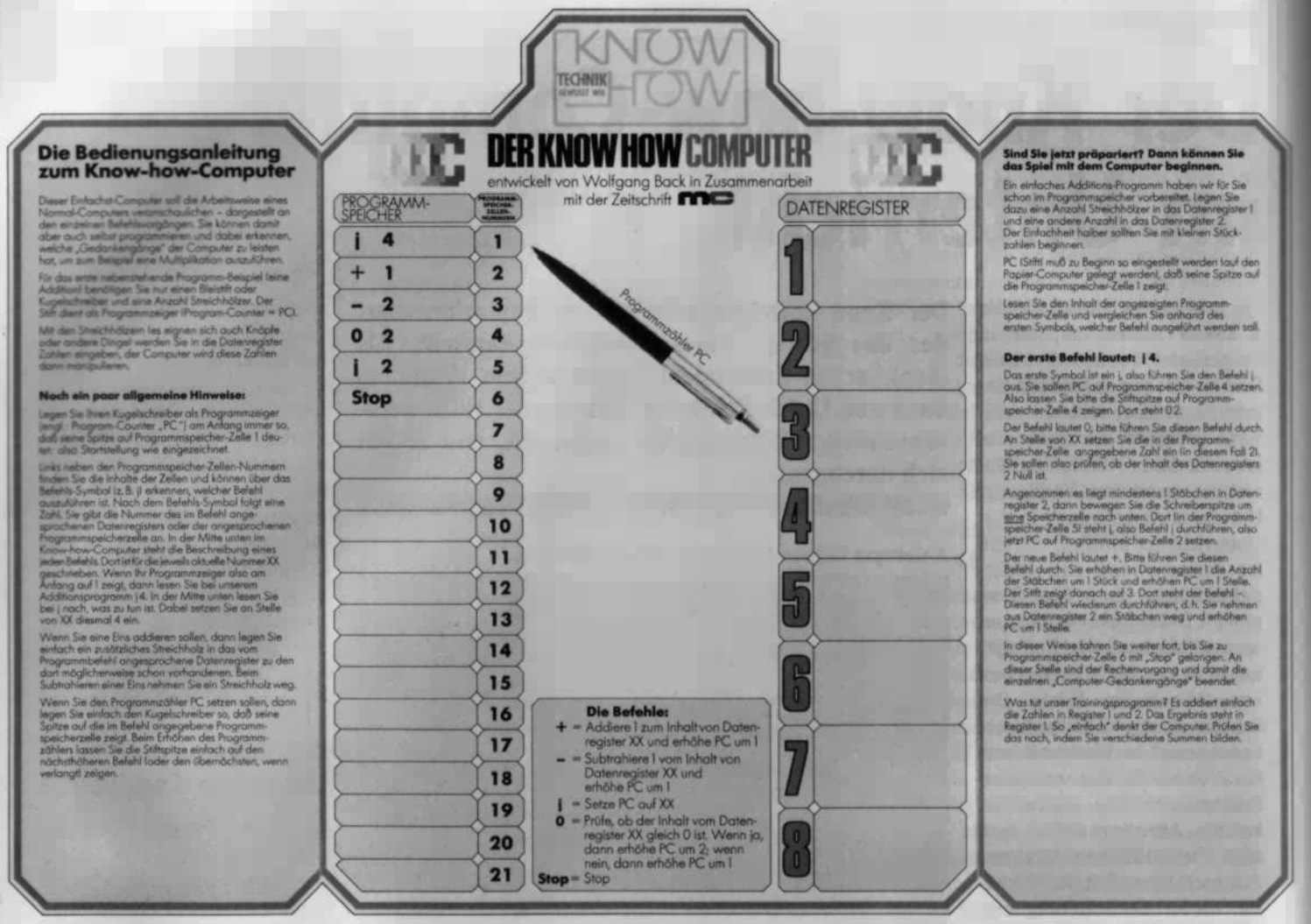


Bild 1. Der Know-how-Computer, hier verkleinert. In natürlicher Größe vom Franzis Verlag zu bekommen

cn
Unterprogrammaufruf, setze Pc auf n+1 und notiere bei 0000=n die Rückkehradresse
stop
Programmende
*
No Operation

In der Tabelle sind die einzelnen Menüpunkte und die Funktion, die sie auslösen, aufgeführt. Die Schüler sollten den vertrauten Papiercomputer in einer Form auf dem PC wiederfinden, in der sich auch viele professionelle Programme dem Benutzer präsentieren.

Aber der Schüler lernt nicht nur den Umgang mit menügeführten Programmen, sondern auch das Thema Papiercomputer stellt sich im neuen Licht: Das am Anfang des Berichtes geschilderte mühsame Radieren und Neuschreiben entfällt. Andererseits ist das Austesten der geschriebenen Programme sehr leicht, weil es die Einzelschrittverarbeitung gibt. Rückkehraddressen werden auf dem Bildschirm negativ dargestellt, so bleibt der Überblick bei Unterprogrammaufrufen gewahrt. Schließlich kann sich der Schüler über Menüpunkt 5 ein detailliertes Protokoll über sämtliche Vorgänge beim Programmablauf machen. *Bild 3* zeigt den Ablauf des Subtraktionsprogrammes aus mc, Ausgabe 6, 1983. Auch kompliziertere Programme sind jetzt machbar. So versuchten sich meine Schüler (Klasse 9), nachdem Addition, Subtraktion und Division relativ mühelos programmiert waren, auch an einem Programm zur Potenzrechnung. Hier kommt dann die Möglichkeit, Unterprogrammtechnik zu realisieren, voll zur Geltung. Bild 2 zeigt das Progamm zur Potenzrechnung. Übrigens fanden nicht nur die Schüler Gefallen am Programm „Know-how-Computer". Auch einige Kollegen waren fasziniert davon, wie man mit wenigen und einfachen Programmbefehlen kompliziertere mathematische Probleme lösen kann und programmierten selbstgestellte Aufgaben. Zwei davon sind in *Bild 2b und 2c* wiedergegeben. *Die Tabelle* zeigt die Anleitung zum Pogramm, so wie sie in der Schule Verwendung findet.

Wie anfangs beschrieben, kam das Programm, bedingt durch die besondere Situation an unserer Schule, in der Klasse 9 zum Einsatz. Komplexere Know-how-Projekte sollten aber den Klassen 10/11 vorbehalten bleiben.

Das Programm „Know-how-Computer" erhalten Sie für einen Unkostenbeitrag von 25 DM auf Diskette (Turbo-Pascal-Quellcode, Beispielprogramme, ComFile lauffähig unter MS-DOS). Den Papiercomputer mit der Anleitung bekommen Sie vom Verlag, wenn Sie einen frankierten Rückumschlag (DIN A4), mit Ihrer eigenen Adresse versehen, an den Franzis-Verlag einsenden und das Stichwort Know-how-Computer dazu angeben. □

a)

```
*> Know How Computer <*

Programm_Speicher                                          Register
  1  -2     16  c61    31>        46>        61>           1  2
  2  j6     17  c31    32  j35    47  j51    62  j65       2  3
  3  -1     18  02     33  -7     48  -1     63  -3        3  8
  4  +3     19  j14    34  c46    49  +3     64  +7        4  0
  5  +8     20  +5     35  07     50  +4     65  03        5  0
  6  01     21  j24    36  j33    51  01     66  j63       6  0
  7  j3     22  -5     37  j31    52  j48    67  j61       7  0
  8  j11    23  +2     38         53  j56    68            8  0
  9  -8     24  05     39         54  +1     69            9
 10  +1     25  j22    40         55  -4     70           10
 11  08     26  stop   41         56  04     71           11
 12  j9     27         42         57  j54    72           12
 13  j18    28         43         58  j46    73           13
 14  -2     29         44         59         74           14
 15  +5     30         45         60         75           15

* 1_Prg_Editor«  2_Reg_Editor«  3_Run_Quick.«  4_Run_Slow..«  5_Exec_Plan.«
  6_Clear_All.«  7_Directory.«  8_Save_Prg..«  9_Load_Prg..«  0_Exit.....«
```

b)

```
Programm_Speicher                                          Register
  1  j4     16         31         46         61            1
  2  -1     17         32         47         62            2
  3  -2     18         33         48         63            3
  4  01     19         34         49         64            4
  5  j7     20         35         50         65            5
  6  j12    21         36         51         66            6
  7  02     22         37         52         67            7
  8  j2     23         38         53         68            8
  9  stop   24         39         54         69            9
 10  +1     25         40         55         70           10
 11  -2     26         41         56         71           11
 12  02     27         42         57         72           12
 13  j10    28         43         58         73           13
 14  +2     29         44         59         74           14
 15  stop   30         45         60         75           15

* 1_Prg_Editor«  2_Reg_Editor«  3_Run_Quick.«  4_Run_Slow..«  5_Exec_Plan.«
  6_Clear_All.«  7_Directory.«  8_Save_Prg..«  9_Load_Prg..«  0_Exit.....«
```

c)

```
Programm_Speicher                                          Register
  1  01     16  -4     31         46         61            1
  2  j4     17  +3     32         47         62            2
  3  stop   18  +5     33         48         63            3
  4  -1     19  04     34         49         64            4
  5  +2     20  j16    35         50         65            5
  6  01     21  -3     36         51         66            6
  7  j4     22  +4     37         52         67            7
  8  03     23  03     38         53         68            8
  9  j11    24  j21    39         54         69            9
 10  stop   25  02     40         55         70           10
 11  -3     26  j15    41         56         71           11
 12  +4     27  stop   42         57         72           12
 13  03     28         43         58         73           13
 14  j11    29         44         59         74           14
 15  -2     30         45         60         75           15

* 1_Prg_Editor«  2_Reg_Editor«  3_Run_Quick.«  4_Run_Slow..«  5_Exec_Plan.«
  6_Clear_All.«  7_Directory.«  8_Save_Prg..«  9_Load_Prg..«  0_Exit.....«
```

Bild 2. a: Aufgeräumt präsentiert sich dem Anwender das Programm Know-how-Com. Hier ist es mit dem Programm „Potenzieren" geladen, Teil b berechnet die Summe der ersten n Quadratzahlen, Teil c das Produkt zweier Summen.

```
                    Programm - Dokumentation
                    ========================

R1 = R1 - R2  (R2 Vorzeichenflag)

Nr  pc  Bef   1  2  3  4  5  6  7  8  9  10 11 12 13 14 15

              2  4  0  0  0  0  0  0  0  0  0  0  0  0  0
 1   1  j4    2  4  0  0  0  0  0  0  0  0  0  0  0  0  0
 2   4  01    2  4  0  0  0  0  0  0  0  0  0  0  0  0  0
 3   5  j7    2  4  0  0  0  0  0  0  0  0  0  0  0  0  0
 4   7  02    2  4  0  0  0  0  0  0  0  0  0  0  0  0  0
 5   8  j2    2  4  0  0  0  0  0  0  0  0  0  0  0  0  0
 6   2  -1    1  4  0  0  0  0  0  0  0  0  0  0  0  0  0
 7   3  -2    1  3  0  0  0  0  0  0  0  0  0  0  0  0  0
 8   4  01    1  3  0  0  0  0  0  0  0  0  0  0  0  0  0
 9   5  j7    1  3  0  0  0  0  0  0  0  0  0  0  0  0  0
10   7  02    1  3  0  0  0  0  0  0  0  0  0  0  0  0  0
11   8  j2    1  3  0  0  0  0  0  0  0  0  0  0  0  0  0
12   2  -1    0  3  0  0  0  0  0  0  0  0  0  0  0  0  0
13   3  -2    0  2  0  0  0  0  0  0  0  0  0  0  0  0  0
14   4  01    0  2  0  0  0  0  0  0  0  0  0  0  0  0  0
15   6  j12   0  2  0  0  0  0  0  0  0  0  0  0  0  0  0
16  12  02    0  2  0  0  0  0  0  0  0  0  0  0  0  0  0
17  13  j10   0  2  0  0  0  0  0  0  0  0  0  0  0  0  0
18  10  +1    1  2  0  0  0  0  0  0  0  0  0  0  0  0  0
19  11  -2    1  1  0  0  0  0  0  0  0  0  0  0  0  0  0
20  12  02    1  1  0  0  0  0  0  0  0  0  0  0  0  0  0
21  13  j10   1  1  0  0  0  0  0  0  0  0  0  0  0  0  0
22  10  +1    2  1  0  0  0  0  0  0  0  0  0  0  0  0  0
23  11  -2    2  0  0  0  0  0  0  0  0  0  0  0  0  0  0
24  12  02    2  0  0  0  0  0  0  0  0  0  0  0  0  0  0
25  14  +2    2  1  0  0  0  0  0  0  0  0  0  0  0  0  0
26  15  stop  2  1  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
*> Know How Computer <*

Programm_Speicher                                          Register
  1  c16    16>        31>        46  j35    61            1
  2  c31    17  01     32  011    47  012    62            2
  3  -1     18  j20    33  j35    48  j50    63            3
  4  01     19  j16    34  j31    49  j31    64            4
  5  j1     20  -1     35  -11    50  -12    65            5
  6  stop   21  +11    36  -12    51  012    66            6
  7         22  +12    37  +13    52  j50    67            7
  8         23  +13    38  +15    53  j31    68            8
  9         24  01     39  012    54         69            9
 10         25  j20    40  j36    55         70           10
 11         26  -13    41  -13    56         71           11
 12         27  +1     42  +12    57         72           12
 13         28  013    43  013    58         73           13
 14         29  j26    44  j41    59         74           14
 15         30  j16    45  011    60         75           15

* 1_Prg_Editor«  2_Reg_Editor«  3_Run_Quick.«  4_Run_Slow..«  5_Exec_Plan.«
  6_Clear_All.«  7_Directory.«  8_Save_Prg..«  9_Load_Prg..«  0_Exit.....«
```

Bild 3. Einzelschrittbetrieb ist möglich. Menüpunkt 5 veranlaßt ein Protokoll im Einzelschrittbetrieb

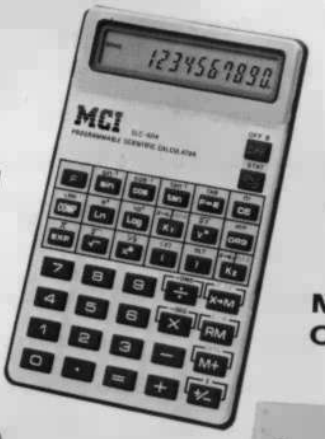
◁

KOMPLETTPREISE...SYSTEMPAKETE... 1 JAHR GARANTIE

... KOMPLETTPREISE...SYSTEMPAKETE

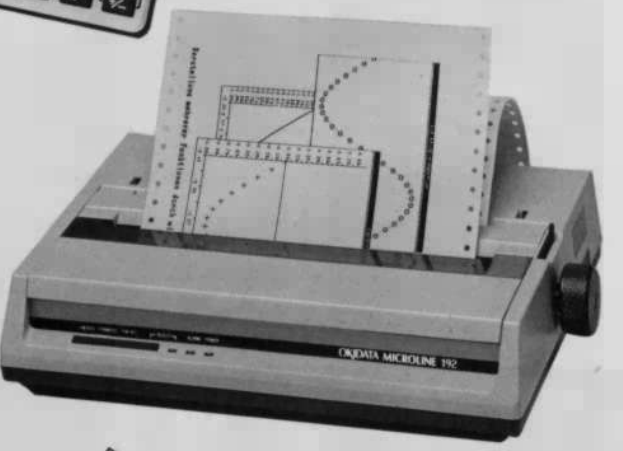
System Pakete für kluge Rechner ab 1349,–

+



(+)

MCI Printer Plus **350,–**
OKI ML192 Elite **899,–**



System Paket 11

10 MHz

1349,–

- **MCI XT16SLC,** 640 K, 1 x 360 K, Clock, ser. par., 12" Monitor, MS-DOS, Tastatur
- **MCI** Programmierbarer Taschenrechner

opt. MCI 120 Zeichen Printer oder OKI ML 192 Elite (Mehrpreis siehe oben)

System Paket 21

10 MHz

1899,–

- **MCI XT16SLC,** 640 K, 1 x 360 K, Clock, ser. par., 12" Monitor, MS-DOS, Tastatur
- **MCI** 20MB Festplatte m. System formatiert
- **MCI** Programmierbarer Taschenrechner

opt. MCI 120 Zeichen Printer oder OKI ML 192 Elite (Mehrpreis siehe oben)

System Paket 31

12 MHz

2399,–

- **MCI AT4SLC,** 640 K, 1 x 1, 2 MB, Clock, ser. par., 12" Monitor, MS-DOS, Tastatur
- **MCI** Programmierbarer Taschenrechner

opt. MCI 120 Zeichen Printer oder OKI ML 192 Elite (Mehrpreis siehe oben)

System Paket 41

12 MHz

3199,–

- **MCI AT4SLC,** 640 K, 1 x 1, 2 MB, Clock ser. par., 12" Monitor, MS-DOS, Tastatur
- **MCI** 20MB Festplatte m. System formatiert
- **MCI** Programmierbarer Taschenrechner

opt. MCI 120 Zeichen Printer oder OKI ML 192 Elite (Mehrpreis siehe oben)

PREIS...QUALITÄT...1 JAHR GARANTIE

KOMPATIBEL...24-STUNDEN-TEST...LEISTUNG...PREIS

MCI XT 16 SLC

alles drin!

Grundausstattung ohne Monitor

ab **899,–**

beinhaltet:

- voll IBM®XT kompatibel
- 8088 CPU + 8087 Sockel
- 8 XT Slots
- 256 KB freier Speicher
- 1 x 360 KB Floppy-Drive
- Color- oder Monochr. Grafikkarte (Hercules II komp. 720 x 348 P.)
- Deutsche Normtastatur MK 5111
- 150 W Schaltnetzteil
- Parallele Drucker-Schnittstelle

Erweiterungen für XT 16 SLC-Serie

2. Laufwerk 360 KB	249,–
Speichererweiterung auf 640 KByte	149,–
Clock/Seriell-Karte	79,–
I/0 Plus II Karte	149,–
20 MB Festplatte mit XT-Controller	+ 589,–
30 MB Festplatte m. RLL XT-Contr.	+ 679,–
EGA-Set statt monochr. Karte	+ 1149,–
Opt. Roll-Maus MO 86 m. Softw.	+ 199,–
Professional Multifunktions-Tastatur MK 6000	+ 100,–
MS-DOS 3.2 + GW-Basic	+ 149,–
9" TTL Monitor grün	+ 150,–
12" Monitor grün od. bern.	+ 229,–
14" TTL Monitor grün, bern. od. weiß	+ 279,–

MCI AT 4 SLC

alles drin!

Grundausstattung ohne Monitor

ab **1899,–**

beinhaltet:

- voll IBM® AT kompatibel
- 80286 CPU + 80287 Sockel
- 6 AT + 2 XT Slots
- 8 und **12 MHz** umschaltbar
- 512 KB freier Speicher
- 1 x 1,2 MB/360 KB Laufwerk
- Color- oder Monochr. Grafikkarte (Hercules II komp. 720 x 348 P.)
- Parallele Drucker-Schnittstelle
- Batteriegep. Echtzeituhr/Kalender
- Kapazitive deutsche Normtastatur

Erweiterungen für AT 4 SLC-Serie

2. Laufwerk 360 KB	299,–
20 MB Festplatte mit AT-Controller	889,–
30 MB Festplatte m. RLL AT-Contr.	+ 999,–
Seriell-Karte	79,–
I/0 Plus II Karte	149,–
EGA-Set statt monochr. Karte	+ 1149,–
MS-DOS 3.2 + GW-Basic	+ 149,–
Professional Multifunktions-Tastatur MK 6000	+ 100,–
9" TTL Monitor grün	+ 150,–
12" Monitor grün od. bern.	+ 229,–
14" TTL Monitor grün, bern./weiß	+ 279,–

PRINTER

MCI *Personal Computer Graphics Printer Plus*

- **voll kompatibel zum** *IBM Personal Computer Graphics Printer*
- **120 Zeichen/sec.**

399,–

OKI MICROLINE ML 192 Elite

- 9 Nadel Matrixdrucker
- Druckgeschwindigkeit 200 Z./sec.
- 40 Zeichen/sec. NLQ
- Druckpuffer 8 KB
- IBM Kompatibel

899,–

EGA

Hochauflösender EGA-Monitor

- Auflösung 320 x 200 (CGA Mode) 640 x 350 (EGA Mode)

EGA-Karte 249,–

999,–

TELEFON-HOTLINE-PREISE
MCI Telefonansage (20 sec.)

	(0 22 02)
Festplatten & Controller	10 81 40
Grafikkarten	10 81 41
Monitore	10 81 42
Schnittstellenkarten	10 81 43
Drucker	10 81 44
Aktuelle Neuigkeiten	10 81 45

MCI

5060 Bergisch Gladbach 2
Bensberger Straße 252
Tel.-Nr.: 02202/1080
Fax: 02202/31009 · Telex: 8873518

12 Monate Garantie auf alle Geräte. Nach der Pang Vo. v. 14. 3. 85 sind wir bei Angeboten gegenüber dem Endverbraucher zur Angabe der **Preise incl. MwSt.** verpflichtet. Für Druckfehler wird nicht gehaftet. **Preise gültig ab 1. 1. 88.** Lieferzeit und Lieferbedingungen auf Anfrage. Änderungen, die technischen Verbesserungen dienen, vorbehalten. Zwischenverkauf vorbehalten.
MCI MICRO COMPUTER INSTRUMENTS GMBH eingetragen AG Bergisch Gladbach · HRB 2575. Herstellung und Vertrieb von Mikrocomputern. 5060 Bergisch Gladbach 2 · Bensberger Straße 252

Horst Brand

Kaum kleiner zu kriegen

Der Aquarius ASI-009, ein XT aus Taiwan mit 3½-Zoll-Diskettenlaufwerken

Bestechend durch seine extrem geringen Abmessungen läßt der ASI-009 außerdem durch einen niedrigen Preis aufhorchen. Da liegt die Befürchtung nahe, daß der Käufer mit Kompromißbereitschaft bezahlen muß, wenn er den Verlockungen dieses Gerätes erliegt.

Die Maße des Winzlings: 26 × 26 × 8 cm^3. Viel kleiner hätte er für die allgemein übliche Aufstellung Monitor-auf-Grundgehäuse auch nicht sein dürfen: der Monitorsockel soll schließlich nicht über den Gehäuserand ragen.

Viel kleiner konnte er bei der horizontalen Anordnung der beiden Laufwerke auch nicht sein: er ist zwar nicht schmaler als zwei, jedoch schmaler als drei direkt nebeneinandergelegte 3½-Zoll-Disketten, und das obwohl sich neben dem rechten Laufwerk noch der Netzkippschalter befindet.

Möglich wurde diese Kompaktheit durch die Verwendung der neuesten Halbleiter, mit denen auf der einzigen Platine die Funktionen der gebräuchlichsten Karten integriert werden konnten: Hercules- und CGA-Farb-Grafik, Floppy-Disk-Controller, zwei serielle Schnittstellen, ein Centronics-Drucker-Port und eine Akku-gepufferte Uhr. Damit ist eine ganze Reihe von Steckplätzen überflüssig geworden; der ASI-009 hat auch nur noch einen Steckplatz, den man zum Beispiel entweder mit einer Netzwerk-, einer Modem-, einer Emulations- oder einer Festplattencontroller-Karte belegen kann. Mit dem Anschluß einer dieser Karten sind aber natürlich die Erweiterungsmöglichkeiten des ASI-009 gänzlich ausgeschöpft.

Bild 1. Fällt neben Monitor und Tastatur kaum noch auf: der ASI-009

Interne Festplatte noch zu teuer

Aufgeräumte Zeiten könnten also mit dem ASI-009 für die Schreibtische aller in Arbeit erstickender Zeitgenossen anbrechen, die für jeden zusätzlich freien Platz in ihrer Büroumgebung dankbar sind. Doch hat die ganze Sache einen Haken. Ein professioneller Einsatz eines Personalcomputers ist mittlerweile ohne Festplatte kaum noch denkbar, und wer einmal mit Festplatte gearbeitet hat, möchte nie wieder auf die ausschließliche Benutzung von Disketten angewiesen sein. Manche Anwenderprogramme sind ja überhaupt nur noch mit Festplatte lauffähig. Das zur Zeit einzige Laufwerk, das in das Gehäuse des ASI-009 paßt, wird von JVC hergestellt und kostet mit Controller ca. 1500 DM (es wird zum Beispiel auch im Laptop Toshiba 1200 eingesetzt). Das wäre jedoch ein krasses Mißverhältnis zum Preis von ca. 850 DM für den ASI-009. Mit einer billigeren externen Lösung (das passende Gehäuse kostet ca. 250 DM) wäre das Bild vom kompakten Billig-PC gleich doppelt demontiert. Eine billigere interne Lösung ist momentan nicht in Sicht und vielleicht in einem halben Jahr zu erwarten, wenn Konkurrenzprodukte zum JVC-Laufwerk auftauchen sollten.

Der Preis von ca. 850 DM gilt für die Version 1, die aus dem Grundgerät mit einem 3½-Zoll-Laufwerk und 256 KByte RAM besteht. Für die Version 2 mit zwei Laufwerken und 640 KByte wird den Händlern kein Verkaufspreis empfohlen. Bei beiden Versionen muß man selbstverständlich die Preise für Tastatur und Monitor hinzurechnen, und dann fällt ein Vergleich mit anderen Angeboten auf dem Markt schon nicht mehr so günstig aus. MS-DOS gehört ebenfalls nicht zum Lieferumfang. Der ASI-009-1 ist nur dann zu empfehlen, wenn man auf absehbare Zeit nur eine Anwendung im Sinn hat und das Programm auch wirklich in den kleinen Arbeitsspeicher paßt. Wer zum Beispiel ausschließlich mit dem Textverarbeitungssystem Euroscript arbeiten will, der kommt bereits mit der Version 1 aus. Die meisten anderen Anwendungen sind allerdings auf einen größeren Speicher angewiesen.

Version 1 läßt sich nachträglich zur Version 2 aufrüsten. Auf der Platine sind bereits die Sokkel für die zusätzlichen RAM-Bausteine vorhanden. Der Bestückung einiger Sockel steht allerdings nach dem problemlosen Öffnen des Gehäuses das Gerüst zur Befestigung der Laufwerksgrundplatte im Wege, so daß man ohne etwas Fingerfertigkeit nicht zum Ziel kommt. Für einen Coprozessor ist ebenfalls ein freier Steckplatz auf der Platine vorhanden. Der ASI-009 arbeitet mit dem Mikroprozessor 8088-1; dieser ist softwarekompatibel zum 8086 und kann mit zwei Taktfrequenzen laufen. Im „Turbo-Mode" beträgt die Taktfrequenz 10 MHz.

Bild 2 zeigt vier Benchmarks für den ASI-009 im Turbo-Mode und für zwei vergleichbare Rechner zur Gegenüberstellung. Die Zeitachse wurde logarithmisch unterteilt, damit alle vier Tests noch gut erkennbar in einer Grafik untergebracht werden konnten. Wenn keine Bildschirmausgabe erfolgt

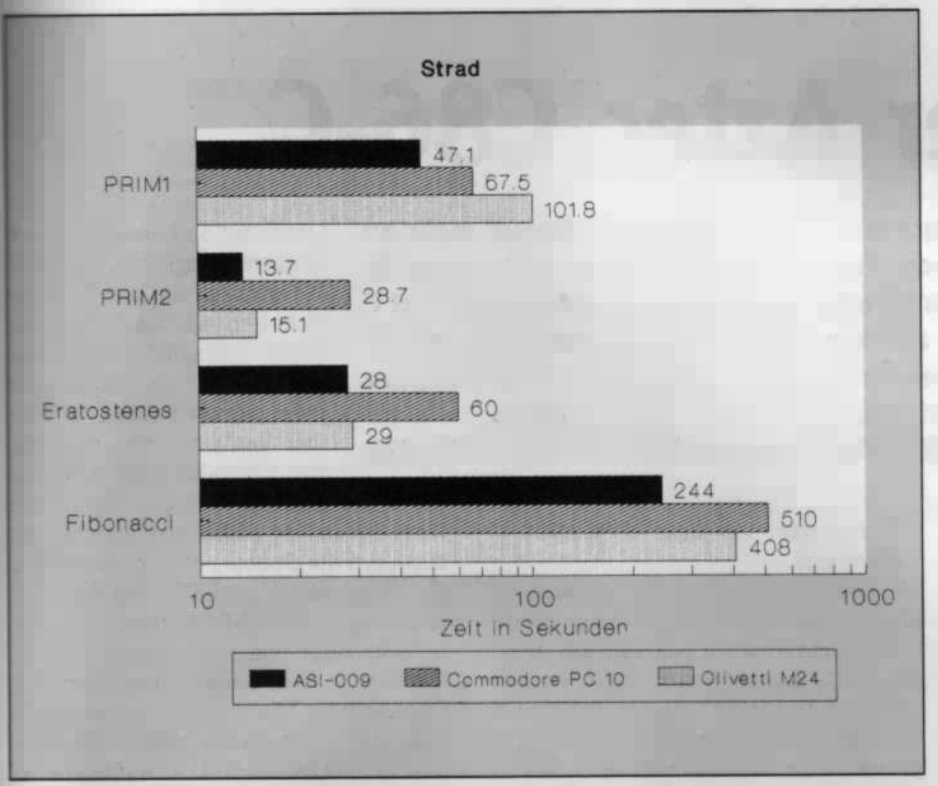


Bild 2. Der ASI-009 und zwei vergleichbare Rechner bei Benchmark-Tests

(Prim2, Fibonacci und Eratosthenes), sind zwischen ASI-009 und PC 10 die Zeitunterschiede exakt auf die unterschiedlichen Taktfrequenzen zurückführbar (10 MHz gegenüber 4,77 MHz); der Olivetti M24 reiht sich leider nicht mit dieser einfachen Erklärung ein. Die Geschwindigkeit der Bildschirmausgabe (bei Prim1 erfolgt eine Bildschirmausgabe) wird von den speziellen Eigenschaften der Grafikkarte bestimmt. Wie man sieht, kommt der ASI-009 bei den Tests recht gut weg.

Testgerät mit kleinen Macken

Die Voreinstellung der Taktfrequenz (beim Einschalten) ist durch eine von insgesamt acht Dip-Schalter-Stellungen im Gehäuseinneren festgelegt. Umschalten zwischen dem Turbo-Mode und der normalen Taktfrequenz 4,77 MHz mit einer Tastenkombination (im Handbuch wird <CTRL><ALT><-> genannt) ist mit der zur Zeit verwendeten BIOS-Version nicht möglich. Die anderen sieben Schiebeschalter des Dip-Blocks geben die Ausbaustufe des ASI-009 und die verwendete Grafikkarte an. Auf der mitgelieferten Diskette befinden sich zwei Programme dhmon und dhcol, mit denen man auch softwaremäßig zwischen den beiden Grafikkarten umschalten können soll. Obwohl beim Einschalten jeweils die Bildschirmdarstellung ohne Beanstandung war, konnten wir nach einmaligem Aufruf von dhmon oder dhcol den alten korrekten Zustand (nach dem Einschalten) mit Hilfe dieser beiden Programme nicht wieder erreichen (wir verwendeten einen MCI-DM14-Monitor). Die Dip-Schalter sind bei geöffnetem Gehäuse mit einem Schraubenzieher gut erreichbar und umlegbar.

Beim Blick ins Innere fiel uns noch etwas auf: Die Versorgungskabel für die Laufwerke sind nicht lang genug und deswegen so gespannt, daß die Stecker schräg auf den Pins der Buchsen sitzen und diese zu verbiegen drohen. Das ist laut Vertreiber S-Trading darauf zurückzuführen, daß ursprünglich Laufwerke verwendet wurden, die ihren Anschluß auf der anderen, für das Kabel näherliegenden Seite hatten, und wurde mittlerweile behoben. Die VDE-, FTZ- und GS-Prüfungen waren zum Zeitpunkt dieses Testes noch im Gange.

Die batteriegepufferte Echtzeituhr muß mit dem Programm Timer auf der mitgelieferten Diskette gestellt werden. Das englische Handbuch umfaßt genau 100 Seiten, und soll vor allem Neulinge auf der Personalcomputer-Szene fürs erste zufriedenstellen. Allein die Erklärung der Tastatur mit der Bedeutung einzelner Tasten für die meisten Anwenderprogramme nimmt gut ein Viertel des Gesamtumfanges ein. Die Besonderheiten des ASI-009 im Vergleich zu Konkurrenzprodukten lassen sich mit Hilfe des Handbuches schnell erkennen. Auf der mitgelieferten Diskette befindet sich auch eine deutsche Übersetzung des Handbuches (man kann es sich kostenlos vom Händler ausdrucken lassen). Wegen der Abbildungen ist das englische Handbuch jedoch nicht entbehrlich.

Während des Testes konnte nicht festgestellt werden, daß es bezüglich der Kompatibilität zum IBM-Standard irgendwelche Probleme gibt.

Das Resümee des professionellen Anwenders wird sicherlich von der internen Verfügbarkeit einer Festplatte stark beeinflußt sein. Was in jedem Fall beeindruckt, ist die faszinierende Kompaktheit dieses XT, und das bei einem niedrigen Preis. Der ASI-009 verrichtet zudem seine Arbeit ohne große Geräuschentwicklung, denn er kommt ohne Ventilator aus.

Spruch des Monats

„Wenn ein Programm erfolgreich ergänzt worden ist, dann will es der Kunde wahrscheinlich nicht haben."

Quelle: Hg. Arthur Bloch: Schiefe Gründe. Warum alles besser geht, wenn es schief geht. Goldmann Verlag, 3. Aufl., 1984.

Thomas Adler, Ingrid Trommer

C-Compiler Aztec C86-C

Die Programmierung in C gewinnt immer mehr an Bedeutung. Früher war C ein reines Werkzeug für Spezialisten unter Unix. Erst im Laufe der Zeit ist C zu einer anerkannten Entwicklungssprache auch für PC-Anwender geworden. Der Aztec C86-Compiler orientiert sich stark an Unix-C-Compilern.

Das Software-Paket Aztec C86-C von Manx Software Systems enthält außer dem eigentlichen C-Compiler noch zahlreiche Utilities. Es bietet eine abgerundete und komplette Palette an Programmen, die man zur Erstellung von C-Programmen unter PC-DOS, MS-DOS und CP/M-86 verwenden kann. In dem beiliegenden Handbuch, das leider nur in Englisch erhältlich und auch etwas unübersichtlich ist, wird versichert, daß die Teile des Sofware-Paketes in Anlehnung an die entsprechenden Werkzeuge des Betriebssystems Unix erstellt wurden. Deswegen wird das Paket mit den entsprechenden Funktionen unter Unix verglichen. Getestet wurde das Programmpaket auf einer Reihe von PCs mit dem Betriebssystem MS-DOS. Der einzige zur Verfügung stehende PC mit dem Betriebssystem CP/M-86 (ein Tele-Video) war leider nicht in der Lage, die Disketten zu lesen. Dies ist nicht weiter verwunderlich, da unter CP/M die Diskettenformate nicht einheitlich sind. Das Programmpaket besteht aus folgenden Teilen:

- einem Text-Editor, der mit dem „vi" unter Unix verwandt ist.
- einem optimierenden, zweiteiligen C-Compiler, der die Syntax überprüft, den Assembler-Code erzeugt und den Assembler aufruft.
- einem Linker, der Objekt-Code-Module zu einem ausführbaren Programm bindet. Bei ausführbaren Programmen unter MS-DOS muß immer ein ".EXE" hinter dem Programmnamen mit eingegeben werden, was für Unix-Anwender etwas gewöhnungsbedürftig ist.
- den wichtigsten Include-Dateien. Z. B. beschreiben die Include-Dateien stdio.h oder math.h. Systemeigenschaften. So steht z. B. in der Include-Datei stdio.h, welchem Code EOF (End of File) entspricht.
- einem Assembler-Debugger und einem Source-Code-Debugger.
- einem Converter, der aus einer Objekt-Datei, die mit dem Aztec-Compiler erzeugt wurde, eine vom Microsoft-Linker verarbeitbare Objekt-Datei macht.
- den Werkzeugen „ls", „diff", „grep", „make".
- Werkzeugen, die unter Unix nicht bekannt sind.

Als erstes wurde der Editor getestet, der hier den Namen Z.EXE trägt. Schon bei den ersten Gehversuchen fällt auf, daß er mit dem „vi", der unter Unix läuft, eng verwandt ist. Die Freude darüber, daß man als Unix-Anwender nicht schon wieder einen neuen Editor kennenlernen muß, läßt einem das Herz gleich höher schlagen. Dennoch stellt man nach einigen Versuchen fest, daß nicht alle Optionen, die der „vi" beherrscht, beim „Z" vorhanden sind. Die Betriebsart, die den „vi" zum Zeileneditor macht, und die solch schöne Dinge wie globale Ersetzungen einer Zeichenkette in der gesamten Datei erlaubt, ist nicht vorhanden.

Leider sind gerade häufig benutzte Funktionen, wie z. B. die Umschaltung vom Einfüge- auf den Kommandomodus, geringfügig verschieden. Bei beiden Systemen wird die Esc-Taste zur Umschaltung verwendet. Auf einer deutschen Tastatur muß man allerdings erst mühsam suchen, unter welcher Bezeichnung diese Funktion versteckt ist. Außerdem muß beim „Z" noch ein zusätzliches Zeichen eingegeben werden, bevor er auf den Kommandmodus umschaltet, da bei diesem Editor das zuletzt eingegebene Zeichen gelöscht wird. Solche Kleinigkeiten fallen einem Unix-Anwender unangenehm auf, wenn er ein C-Programm auf einem MS-DOS-System schreiben will. Die nächste Schwierigkeit tritt beim Suchen der geschweiften Klammern auf der deutschen PC-Tastatur auf. In C sind die geschweiften Klammern so wichtig wie in Pascal die Begin- und die End-Anweisungen.

Der kleine Unterschied

Als Unix-Anwender ist man bei C-Compilern gewöhnt, mit dem Befehl „cc" ein ausführbares Programm zu erhalten. Unix ruft zuerst den C-Compiler und danach den Linker auf und erstellt ein ausführbares Programm. Eventuelle Optionen, die der Compiler nicht versteht, werden an den Linker übergeben. Ruft man hier „cc" auf, so entsteht eine Datei mit der Endung ".o". Erster Verdacht war, daß man dem Compiler sagen muß, daß unter MS-DOS eine ausführbare Datei die Endung „.EXE" hat. Man benutzt deswegen die Option „-o" und schon heißt die Ausgabedatei „xx.EXE". Die Exe-Datei kann man unter MS-DOS starten und wartet sehr lange und sehr gespannt auf ein Ergebnis. Je nach Geduld bootet man nach längerer oder kürzerer Zeit dann das System neu. Irgendwann stellte sich dann ganz zufällig heraus, daß der Aufruf von „c" die obigen Anforderungen erfüllt. Aus der Dokumentation war das allerdings nicht ersichtlich.

Der Linker des Programmpakets ist seiner Aufgabe gut gewachsen. Bis auf die Tatsache, daß zusätzlich zum ausführbaren Programm ein „.EXE" mit dem Pogrammnamen eingegeben werden muß, gibt es gegenüber dem Unix-System keine nennenswerten bzw. sichtbaren Unterschiede. Die Anzahl der Include-Dateien ist wesentlich geringer als bei einem Unix-System. Das Verhältnis der Anzahl der Include-Dateien unter dem Unix-C-Compiler und dem Aztek-C-Compiler beträgt 89 zu 24.

Während des Testes erschien mir der Source-Code-Debugger als ein „verkappter" Assembler-Debugger. Es mag auch an der ungenügenden Beschreibung gelegen haben, denn es war z. B. nicht möglich, den Source-Code-Debugger dazu zu bewegen, bei einem Befehl wie „Single-Step" auch wirklich C-Source-Code anzuzeigen.

Eine weitere Utility – der Konverter – ist für MS-DOS-Rechner vorgesehen, die nicht ganz IBM-kompatibel sind. Dieses Werkzeug wäre auch für andere Programmpakete wünschenswert. Die Werkzeuge „diff", „grep" und „make" haben die gleiche Funktion, wie die gleichnamigen unter Unix. Diese wurden in der Serie „Einführung in Unix" [1] beschrieben. Die Optionen der Werkzeuge sind zwar nicht so umfangreich wie unter Unix, aber sie lassen sich dennoch gut einsetzen. Positiv ist, daß die Utility „make" dem Unix-Anwender sehr entgegenkommt. Allerdings muß man auch an-

```
/*sorting benchmark */
#define MAXNUM 1000
#define COUNT 100
#define MODULUS ((long ) 0x20000)

#define C 13849L
#define A 25173L
long seed = 7L;
long random();
long buffer [MAXNUM] = {0};

main()
{
        int i,j;
        long temp;

        printf("Filling array and sorting %d times \n", COUNT);
        for (i = 0; i < COUNT; ++i) {
            for (j = 0; j < MAXNUM; ++j) {
                temp = random (MODULUS);
                if ( temp < 0L)
                   temp = (-temp);
                   buffer[j] = temp;
            }
            printf ("Hallo World! iterations %d \n",i);
            quick (0, MAXNUM -1, buffer);

        }
}


quick(lo,hi,base)
        int lo, hi;
        long base [];
        {
        int i,j;
        long pivot, temp;

        if (lo < hi)
        {
        for (i = lo, j = hi - 1, pivot = base[hi]; i < j;)
        {
        while ( i < hi && base [i] <= pivot)
              ++i;
        while ( j > lo && base [j] >= pivot)
              --j;
        if (i < j)
           {
           temp = base[i];
        if (i < j)
           {
           temp = base[i];
           base [i] = base[j];
           base [j] = temp;
           }
        }
        temp = base[i];
        base [i] = base[hi];
        base[hi] = temp;
        quick(lo, i-1, base);
        quick(i+1, hi, base);
        }
}

long random (size)
        long size;
        {
        seed = seed*A+C;
        return (seed % size);
        }
```

merken, daß die Anzahl der vorhandenen Werkzeuge gegenüber Unix doch etwas gering ist. Es gibt eineinhalb Programme, die gerade für den unerfahrenen C-Programmierer eine große Hilfe sein könnten. Es sind „C-nach-Englisch" und „Englisch-nach-C". Man kann hier komplizierte C-Deklarationen eingeben und erhält eine englische Erklärung dafür (C-nach-Englisch). Der umgekehrte Weg erscheint fragwürdig, da man, um die richtige Syntax zu erhalten, einen entsprechenden genauen Wortlaut eingeben muß, der nicht immer bekannt ist.

Die übliche Methode, Leistungen von Compilern zu testen, sind „Benchmark-Programme". Sie sollen die unterschiedlichen Fähigkeiten eines Compilers in bezug auf Rechengeschwindigkeit von Integer- und Fließkomma-Zahlen oder der Bibliotheksfunktionen ermitteln. Der Quellcode des Testprogramms ist in *Bild 1* dargestellt. Mit ihm wird eine nahezu 100prozentige CPU-Auslastung erzielt. Der Test wurde auf einem AT von Tandon (80286-CPU mit 8 MHz) durchgeführt und anschließend mit einem C-Compiler unter Unix auf einer HP-Maschine (HP 9000 ohne Coprozessor) verglichen. Der Vergleich der beiden Compiler fiel unentschieden aus. Auf beiden Rechnern betrug die Rechenzeit 52 s. Allerdings benötigt der Unix-Compiler zum Compilieren weniger als die halbe Zeit wie der Aztec-Compiler (9,7 s zu 24,8 s).

Für Unix-Umsteiger geeignet

Die Programmiersprache C ist unter Unix gewachsen und war von Anfang an für Programmentwickler und Techniker konzipiert. Es wird demnach Zeit diese Sprache auch anderen Betriebssystemen zugänglich zu machen. Der Aztek C-Compiler ist für MS-DOS-Anwender ein guter Einstieg in die Programmiersprache C. Die Nähe und die Ähnlichkeit zu den Werkzeugen unter Unix versetzt sogar den fortgeschrittenen Unix-Anwender in die Lage auf anderen Betriebssystemen Programme zu schreiben. Selbst wenn sich das Programmpaket in einigen Punkten hinsichtlich des Komforts und der Übersichtlichkeit von C-Compilern unter Unix unterscheidet.

Literatur

[1] Trommer, Ingrid: Einführung in Unix. mc 1987, Ausgabe 6...12.

Das Quicksort-Programm ist in C geschrieben

Reiner Schönrock

Shell mit Editor für Atari ST

Mit Craft geht's auch ohne GEM

Bei der Arbeit mit dem Atari ST unter GEM vermissen viele professionelle Umsteiger einige Befehle. Außerdem dauert der Bildaufbau mit dem Zeichnen aller Desktop-Elemente und dem immer wieder neuen Einlesen der Inhaltsverzeichnisse lange. Eine neue kommandoorientierte Shell ist besser.

Aus Amsterdam kommt ein interessantes Programm für den Atari ST. Das hierzulande noch weitgehend unbekannte Softwarehaus Commedia bringt mit Craft einen General Purpose Command Interpreter mit einem Editor auf den Markt. Besonders an Programmierer wendet sich das Produkt, das sich aus einer Diskette und einem 500 Seiten dicken Handbuch zusammensetzt. Auf der einseitig formatierten Diskette befinden sich in zehn Unterverzeichnissen mit insgesamt 58 Dateien zahlreiche Utilities, die die Arbeit mit dem Atari ST an Software-Projekten vereinfachen und beschleunigen.

Viele nützliche Utilities

Die GPShell ist kommandoorientiert und kennt 52 Kommandos, die im wesentlichen an Unix angelehnt sind. Diese Kommandos werden unmittelbar von der Shell aus gestartet. Das Shell-Programm selbst ist etwa 65 KByte groß. Die weiteren externen Kommandos werden von der Festplatte oder Diskette geladen.

Beim Starten der Shell sucht das Programm die Datei „Login.sh", die ähnlich der Datei „Autoexec.bat" unter MS-DOS Befehlsfolgen enthält. Damit werden Grundeinstellungen für z. B. das Stammverzeichnis, die Cursor-Blinkrate, die Monitor-Darstellung, den Font und beliebige andere Systemeinstellungen vorgenommen. Die Shell bietet unter anderem das Kommando „Ramdisk", mit der eine RAM-Disk beliebiger Größe eingerichtet wird. Bindet man dieses Kommando in die Datei „Login.sh" ein, hat man während der Arbeit mit der Shell eine RAM-Disk, die beim Verlassen der Shell wieder aus dem System entfernt wird. Damit man nicht aus Versehen von der Shell aus zu GEM wechselt und vielleicht dabei Daten von der RAM-Disk verliert, gibt die Shell eine Warnung aus.

Um die Arbeit am Atari ST weiter zu beschleunigen, könnte man das ebenfalls zu Craft gehörende Programm „Fastdisk" in den Auto-Ordner der Start-Diskette kopieren. Das nur wenige Byte umfassende Programm stellt die Seekrate auf 2 ms ein. Die Seekrate gibt an, wieviel Zeit der Schreib-/Lesekopf der Floppy-Laufwerke hat, um von einem Track zum nächsten zu wechseln. Sie wird vom TOS aus standardmäßig auf 3 ms eingestellt. Da nach dem Lauf des Programmes „Fastdisk" jeder Track-Wechsel 1 ms schneller abläuft als vorher, bemerkt man die Beschleunigung wahrscheinlich garnicht. Dieses Programm erinnerte mich an die Additive für Benzin oder Diesel. Ob die nun wirken weiß man als Autofahrer auch nicht – aber es könnte ja sein. Bei meinen Doppel-Floppy-Laufwerken von Cumana traten nach der Verkürzung der Seekrate sehr häufig Schreib-/Lesefehler auf dem Laufwerk „B" auf. Durch eine Änderung der Seekrate auf den Standardwert funktionierte wieder alles normal. Um der Datensicherheit Willen sollte man sich etwas mehr Zeit nehmen und das Programm „Fastdisk" aus dem Auto-Ordner herauslassen.

Sinnvoller ist da schon das ebenfalls beiliegende Programm „Loader". Das Betriebssystem des Atari ST schreibt vor dem Start eines jeden Programmes in jede freie Speicherzelle eine Null. Je mehr Speicherplatz zur Verfügung steht, umso länger dauert dieser Vorgang. Dadurch entstehen auch die merklich längeren Wartezeiten bei den neuen Mega ST-Computern. Das Löschen von 1 MByte RAM schafft das TOS in etwas mehr als zwei Sekunden. Für 2,5 MByte werden über fünf Sekunden gebraucht. Mit dem Programm „Loader" werden je MByte RAM nur noch etwa 0,3 Sekunden zum Rücksetzen des freien Speichers benötigt. Umso größer der zur Verfügung stehende Speicher in einem Atari ST ist, umso deutlicher macht sich die neue Lade-Routine bemerkbar. Braucht z. B. ein C-Compiler auf einem Atari ST mit 2,5 MByte RAM zur Compilierung eines Programmes über 40 Sekunden, so kann der Loader eine Zeitersparnis von fast 20 Sekunden bringen. Dieses Programm ist sinnvoll und muß im Auto-Ordner als erstes auszuführendes Programm installiert werden. Das „Loader"-Programm installiert auch noch einen neuen GEMDOS-Befehl, der dazu benutzt werden kann, daß nur das BSS-Segment eines Programmes gelöscht wird. Dadurch läßt sich eine weitere Beschleunigung erzielen.

Das „Shell"-Programm selbst kann auch in einen AUTO-Ordner kopiert werden. Beim Booten des Atari ST wird sie dann sofort gestartet. Bei der Arbeit mit der Shell ist das nicht weiter schlimm, denn die Shell ist rein kommandoorientiert und stört sich nicht an der Maus. Anders sieht es da schon bei der Arbeit mit dem zu Craft gehörenden Editor Stedi aus. Dieser kann nämlich sowohl mit der Maus als auch mit Kommandos bedient werden. Ganz besonders wichtig ist aber, daß man die Shell von GEM aus bootet, wenn man vom Editor aus GEM-Programme aufrufen möchte. Die Entwickler von Craft haben diesem Umstand Rechnung getragen und liefern ein Autostart-Programm mit. Damit wird die Shell erst nach der Initialisierung der AES-Treiber gestartet. Sie gehen dabei einen eigenwilligen aber flexiblen Weg. Nach dem Booten wird nämlich das Programm gestartet, über dessen Eintrag sich der Mauszeiger befindet. Da der Mauszeiger sich nach dem Einschalten oder einem Reset immer genau in der Mitte des Bildschirmes meldet, muß man nur dafür sorgen, daß der Name des zu startenden Programmes genau in der Mitte des Bildschirmes zu liegen kommt. Zur Positionierung hilft ein kleiner Klecks Wasserfarbe auf der Mattscheibe, der die Startposition des Mauszeigers anzeigt. Wer also gerne sofort nach dem Einschalten des Atari ST mit seinem Lieblingsprogramm arbeiten möchte, kann das leicht tun. Dem Willen der Craft-Entwickler nach sollte natürlich das Shell-Programm in der Mitte des Monitors liegen und sofort gestartet werden.

Der größte Vorteil der Kommando-Shell im Vergleich zu GEM ist, daß man flexibler, schneller und einfacher mit dem Computer kommunizieren kann. Das sei an einigen Beispielen demonstriert. Wenn z. B. in einem Unterverzeichnis einer Festplatte über

200 Dateien stehen, dauert es je nach Aufbau des Desktop sehr lange, bis man die gewünschte Datei unter dem Mauszeiger hat. Nach jeder Betätigung der Rollbalken muß man einige Zeit auf den neuen Aufbau der Inhaltsverzeichnisse warten. Mit der GPShell gestaltet sich die Arbeit einfacher. Hier muß man nur den Anfangsbuchstaben der gewünschten Datei eingeben und die ESC-Taste drücken. Die Shell ergänzt den passenden Namen. Diesen kann man durch „Return" übernehmen. Ein weiterer Vorteil der GPShell ist das Kopieren von ausgewählten Dateien. Unter GEM muß man zuerst die Art der Sortierung aller Dateien in einem Fenster festlegen, um dann z. B. alle Dateien mit der Extension „DAT" kopieren zu können. Mit der GPShell beschränkt sich der Arbeitsaufwand auf die Befehlsfolge „cp *.dat Ziel:", worauf alle zutreffenden Dateien – in diesem Beispiel die mit der Extension „dat" – kopiert werden. Wer auf dem Monitor seines Atari ST vier Fenster mit Inhaltsangaben von Disketten hält, muß sich nach einem Diskettenwechsel oder nach dem Beenden eines Programmes in Geduld fassen. Bis alle Verzeichnisse gelesen und neu gezeichnet sind, braucht es schon ein paar Sekunden. Bei der GPShell entfällt der Service der allzeit präsenten Inhaltsverzeichnisse. Hier meldet sich nur ein schlichter Prompt, der auf die Eingabe von Befehlen wartet. Dieser Prompt ist standardmäßig ein „$", kann aber beliebig geändert werden. Weitere Vorteile der GPShell sind Redirection, Aliasing, Pipes, frei definierbare Funktionstasten, History-Liste und weitere mehr.
Die Befehlszeile der GPShell kann soviele Befehle aufnehmen, wie auf den Bildschirm passen. An Programme können so beliebig viele Parameter übergeben werden. Möchte man z. B. den zur GPShell gehörigen Editor Stedi starten und in mehreren Dateien Änderungen vornehmen, kann man bis zu sechs Dateinamen angeben.

Zwei Betriebsarten möglich

Die GPShell kann in zwei „Betriebsarten" benutzt werden, im GEMDOS- und im Unix-Mode. Der Unterschied zwischen den beiden Arten liegt hauptsächlich in der Darstellung der Directory-Separatoren. Aus dem „/" wird ein „\". Auch die Funktion der ESC-Taste ändert sich. An den Kommandos und deren Funktionen ändert sich dadurch nichts.
In *Bild 1* sind alle Kommandos mit einer Kurzbeschreibung zusammengefaßt dargestellt. Besonders interessant ist nicht nur die hohe Anzahl von Kommandos sondern auch, daß die Befehlszeilen frei editiert

Name der internen Funktion	Wirkung
alias	Umbenennung von Funktionsnamen. Nach „alias h 'history'" wird z. B. die History-Liste durch Eingabe von „h" angezeigt
alloc	Informationen über die Shell-Heap-Parameter. Anzeige der belegten und der freien Speicherbereiche
bell	Ein-/Ausschalten der Glocke
cat	Ausgabe von Dateien auf das Standard-Ausgabegerät. Ähnlich „TYPE" von MS-DOS. Mit Redirection Ausgabe in Dateien
cd	Wechsel der Inhaltsverzeichnisse
chmod	Änderung der Datei-Attribute
clear	Löscht den Bildschirm
cp	Kopiert Dateien und Inhaltsverzeichnisse
cursor	Stellt die Blinkrate des Cursors ein
date	Zeigt Datum und Zeit an. Bei Angabe von Parametern werden diese übernommen
df	Zeigt den freien Speicher der Massenspeicher an
dirs	Zeigt den Directory-Stack an
echo	Ausgabe von Zeichenketten
errorfile	Legt eine Fehler-Datei an
find	Sucht auf dem angegebenen Datenträger nach Dateien, die bestimmten Kriterien entsprechen; find c: 't*.*' z. B. zeigt alle Dateien der Festplatte C an, die mit einem „t" anfangen
font	Wählt Zeichenfont aus
free	Zeigt den freien Speicher an
hashtable	Zeigt die verfügbaren externen Kommandos an
hashstat	Gibt Auskunft über die Effektivität, mit der externe Kommandos gesucht wurden
help	Gibt Hinweise zur Bedienung
history	Anzeige der letzten X Befehle
keyclick	Schaltet Tastatur-Klick an oder aus
keydelay	Stellt die Zeit ein, nach der die Repeat-Funktion einsetzt
keyrepeat	Stellt die Geschwindigkeit der Repeat-Funktion ein
logout	Verläßt die Shell
lpr	Gibt Datei auf Drucker aus
ls	Anzeige des Inhaltsverzeichnisses. Sieben Parameter zur Selektion
man	Zur Anzeige umfangreicher Hilfstexte
mkdir	Legt neues Unterverzeichnis an
moreheap	Vergrößert den Shell-Heap
mount	Meldet Laufwerke an
mv	Verlegt Dateien in andere Verzeichnisse oder benennt sie neu
palette	Stellt die Bildschirmfarbe ein
popd	Wechselt das Verzeichnis nach dem Directory-Stack
printenv	Zeigt Environment-Variable an
pushd	Legt den Namen eines Verzeichnisses auf den Directory-Stack
pwd	Zeigt aktuelles Verzeichnis an
ramdisk	Legt RAM-Disk an
read	Belegt Shell-Variable
rehash	Reorganisation der Hashtable
rm	Löscht Dateien
rmdir	Löscht leeres Unterverzeichnis
set	Belegt Shell-Variable oder zeigt die Belegung an
setenv	Definiert eine Environment-Variable
sleep	Legt den Computer für eine definierte Zeit lahm
touch	Ändert die Datum-/Zeit-Informationen von Dateien
umount	Meldet Laufwerke ab
unalias	Hebt Umbenennung eines Kommandos auf
unset	Löscht eine Shell-Variable
unsetenv	Löscht eine Environment-Variable
verify	Schaltet das Verify-Flag
which	Zeigt Dateien an, die als Kommandos erkannt werden
who	Zeigt den Namen des Anwenders und die Zeit des Login an

Name der externen Funktion	Wirkung
cmp	Vergleicht zwei Dateien
du	Zeigt die Belegung von Disketten oder Verzeichnissen an
dump	Zeigt Dateien hexadezimal und in ASCII an
envtest	Dient zum Test der Variablen der Environment-Liste
fastdisk	Setzt die Seek-Rate auf 2 ms
format	Formatiert Disketten
getpr	Zeigt die Druckerkonfiguration an
loader	Neue Lade-Routine für Programme
lpq	Zeigt den Inhalt der Drucker-Warteschlange an
make	Make-Utility
more	Ausgabe von Dateien am Bildschirm stoppt nach einer Seite
normdisk	Hebt Fastdisk auf
od	Anzeige von Dateien als hexal, oktal oder ASCII-Dump
pr	Formatiert Dateien und zeigt sie am Bildschirm an. Einstellung mit sechs Parametern
setpr	Stellt den Drucker ein
shell	Startet von der Shell aus eine neue Shell. Kontrolle über neun Parameter
size	Gibt Auskunft über die Text-, Data-, BSS- und Smybol-Segmente von Programmen
wc	Zeigt Anzahl der Zeilen, Zeichen und Wörter einer Datei an

Bild 1. Alle Kommandos der GPShell von Craft

werden können. In einer History-Liste werden alle Kommandos in der Reihenfolge, in der sie benutzt werden, gespeichert. In der Praxis hat sich gezeigt, daß es ausreichend ist, die jeweils letzten 20 Kommandos in dieser Liste zu behalten. Mit dem Kommando „history" erhält man dann eine Auflistung der z. B. letzten 20 Befehle und kann durch "!xx" das entsprechende Kommando aufrufen. Dieses Utility spart einiges an Tipparbeit.

Damit man von selbstgeschriebenen Programmen aus auf die Shell-Kommandos zugreifen kann, wird die Variable „_shell_p" initialisiert. Von C-Programmen aus wird diese Variable als Zeiger auf eine Funktion eingesetzt. Wie das im einzelnen aussieht zeigen die Entwickler von Craft an zahlreichen Beispielen für die gängigsten C-Compiler. Der Source-Code ist auf der Programm-Diskette enthalten.

Neben der Shell, die jeder Umsteiger von einem MS-DOS-Computer oder einem anderen kommandoorientierten System zu schätzen wissen wird, bietet Craft noch einige Besonderheiten. Da sind das RAM-Disk-Kommando und das Loader-Programm. Aber auch ein Drucker-Spooler, ein Make-Utility und die Input-/Output-Redirection helfen bei der Programm-Entwicklung mit dem Atari ST.

Ein weiterer wichtiger Bestandteil von Craft ist der Editor Stedi. Der Editor ist auf drei Arten zu erreichen: von der GPShell aus, von GEM aus als Programm und als Accessory. Der Aufruf von der GPShell aus hat den Vorteil, daß man alle Shell-Kommandos und GEM-Programme vom Editor aus aufrufen kann. Dieses funktioniert beim Aufruf als GEM-Programm oder als Accessory nicht.

Als erstes erscheint nach dem Start die Datei-Auswahlbox von GEM. Sie sieht zwar aus wie die Standard-Box von GEM, unterscheidet sich aber durch eine Autoscroll-Funktion bei dem Rollbalken zur Auswahl der Dateien.

Der Editor wurde komplett in Assembler programmiert und ist entsprechend schnell. Die Bedienung der zahlreichen Funktionen kann wahlweise mit der Maus, mit den doppelt belegten Funktionstasten oder über Tastatur-Kommandos (ALT- und CTRL-Taste) vorgenommen werden. Bis man den Editor richtig beherrscht, muß man einige Zeit lang damit üben und die wirklich gute und umfangreiche Dokumentation durcharbeiten. Das Programm ist längst nicht so einfach zu bedienen wie Tempus oder 1ST-Word+. Wer lieber mit der Maus arbeitet, muß bei Stedi die Kommandos aus der untersten Bildschirmzeile herauspicken. Hier werden nur Anfangbuchstaben vorgegeben, z. B. „R" für „Read File". Die von GEM her gewohnten Pull-Down-Menüs gibt es bei Stedi nicht. In der untersten Bildschirmzeile werden außer den mit der Maus auszuwählenden Kommandos auch Meldungen ausgegeben. GEM-Programme öffnen meist Fenster mitten im Bildschirm, wenn sie etwas zu melden haben. Zur ersten Orientierung über den Funktionsumfang von Stedi mag ein Druck auf die „HELP"-Taste genügen. Daraufhin werden etwa 114 Zeilen – das entspricht fünf Seiten Text – eingeblendet, die alle Funktionen zusammenfassend darstellen.

Zitat des Monats

Bei einem Strom- oder Computerausfall erscheint gegebenenfalls die Meldung „Sind andere Kopien des Programm-Editors aktiv? (J/N)" bzw. „Sind andere Kopien des Macro-Editors aktiv? (J/N)".

Aus einer Bedienungsanleitung

Mit Stedi kann man acht Texte gleichzeitig bearbeiten. Zwei weitere Puffer stehen für Cut- und Paste-Funktionen zur Verfügung. Auf dem Monitor ist dabei aber immer nur ein Text sichtbar; beliebig verschieb- oder änderbare Fenster gibt es bei Stedi nicht. Eine Zeile darf maximal 255 Zeichen, ein Text darf maximal 99999 Zeilen lang sein. Zum Speichern von Dateien bietet Stedi vier Modi an: Atari-, Unix-, Raw- oder Print-Mode. Im Atari-Modus wird jede Zeile mit Carriage Return und Linefeed gespeichert. Beim Unix-Mode wird zu jeder Zeile nur ein Linefeed gespeichert. Wer eine Datei „so wie sie ist" speichern möchte, muß den Raw-Mode wählen. Dann werden weder Carriage Return noch Linefeed gespeichert. Dieser Modus empfiehlt sich für binäre Dateien. Da Stedi auch die Eingabe von Zeichen über den hexadezimalen Code erlaubt, hat dieser Modus seine Berechtigung. Durch den Print-Mode werden schließlich alle Tabs durch Leerzeichen ersetzt.

Wir wollen an dieser Stelle nicht auf die Funktionen eingehen, die eh jeder Editor bietet. Stedi hat einiges in Petto, was man sonst nicht geboten bekommt. Aus dieser Auswahl ist für größere Softwareprojekte besonders das Strukturelement Folder wichtig. Bevor man sich daran macht, ein umfangreiches Programm-Listing einzutippen, gibt man in Stedi zuerst die Namen der einzelnen Routinen ein – sofern man strukturiert programmiert. Danach beginnt man mit dem Eintippen der einzelnen Routinen. Hat man eine Routine fertig, genügt ein Druck auf eine Funktionstaste, und von der Routine bleibt nur die Kopfzeile auf dem Monitor sichtbar. So verliert man selbst bei den größten Programmen nicht die Übersicht, da auf Wunsch nur die Modul-Köpfe angezeigt werden.

Überaus nützlich sind auch die zehn frei programmierbaren Tastatur-Makros. Jedes Makro kann 100 Tastendrücke speichern. Alle Funktionen können dabei benutzt werden. In einer speziellen Datei, die alle Parameter speichert, werden auch die Tastaturmakros gespeichert.

Eine weitere Besonderheit ist die vollständige Programmierbarkeit der Tastatur. Das dazu erforderliche Programm ist mit dem C-Sourcecode auf der Diskette enthalten. Vom Editor aus kann man mit Kommandos den Drucker einstellen, Dateien drucken, den freien Speicher von Disketten oder Festplatten feststellen und natürlich auch Suchen und Ersetzen. Bei letzterem ist die Richtung in der Kommandozeile einstellbar.

Der Editor Stedi dient dem Bearbeiten von Textdateien. Die Funktionen, die aus diesem Programm ein Textverarbeitungsprogramm machen könnten, werden in absehbarer Zeit nicht implementiert. Um einen Text zu formatieren bietet Stedi eine Word-Wrap-Funktion. Damit stellt man den rechten Rand ein, an dem das Programm den eingegebenen Text umbricht. Fonts können jederzeit geladen werden. Eine letzte interessante Funktion die erwähnenswert ist heißt „auto-indent". Damit wird eine frei definierbare Zeichenfolge bei jeder Zeilenschaltung an den Anfang der folgenden Zeile gesetzt.

Im Vergleich zu anderen Editoren für den Atari ST steht Stedi sehr gut da. Von der Geschwindigkeit her kann er es mit dem Programm Tempus aufnehmen. Von der Bedienung her, die über ALT-, CTRL- und Funktionstasten lernintensiv ist, fehlt ein vergleichbares Produkt. Die meisten Editoren für den Atari ST wurden in GEM eingebunden oder bieten nicht die Leistungsfähigkeit von Stedi. Die frei definierbaren Tastatur-Makros, das Folder-Kommando zur Strukturierung umfangreicher Texte und die Schnittstelle zur GPShell zeichnen den Editor aus.

Wer auf dem Atari ST an umfangreichen Software-Projekten arbeitet, dabei aber gerne auf den GEM-Overhead verzichten möchte, ist mit Craft bestens bedient.

Für eine Diskette und ein 500 Seiten dickes englisches Handbuch muß man um 300 DM zahlen. □

Die Lösung am Host

QMS-*LASERGRAFIX* 1510

Mit einem Lasergrafix von QMS erstellt Ihr Host sekundenschnell Vektorgrafik, Balkengrafik, Barcodes, Formulare und Texte in den unterschiedlichsten Schriften. 15 Seiten DIN A4 sind es, die der Lasergrafix 1510 in einer Minute zu Papier bringt. Ebenso schnell schafft sein kleiner Bruder LG 800II 8 Seiten, der größte der Familie LG 2400 sogar 24 Seiten in der gleichen Zeit – ganz gleich ob in DIN A4 oder DIN A3. Alle 3 sind durch die Seitenbeschreibungssprache QUIC kompatibel zu einer breiten Softwarepalette wie z.B. CA (ISSCO), SAS, Medusa und PVI. Durch DCF+ lassen sich alle QMS Lasergrafix-Modelle voll in die IBM-Host-Umgebung einbinden.

PRINT PARTNER

Die Drucker-Spezialisten

Berlin
INFORDATA Datentechnik GmbH Berlin
Eisenacher Straße 123, 1000 Berlin 30
Tel. (030) 216 50 50

Hamburg, Bremen, Schleswig-Holstein
TEXTCOM Mikrocomputer-Vertrieb GmbH
Steindamm 32, 2000 Hamburg 1
Tel. (040) 24 36 16, Telex 214768

Niedersachsen
TEXTCOM Mikrocomputer-Vertrieb GmbH
Geschäftsstelle Hannover
Landwehrstraße 61, 3000 Hannover 81
Tel. (05 11) 83 09 00

Nordrhein-Westfalen
AC-COPY Datentechnikvertrieb GmbH
Peliserker Straße 69, 5100 Aachen
Tel. (02 41) 50 60 96, Telex 832368
Geschäftsstelle Dortmund
Baroper Straße 261, 4600 Dortmund 50

Hessen, Rheinland-Pfalz, Saarland
PRINTEC Datentechnik Vertriebs-GmbH
Dieselstraße 20,
6452 Hainburg 2
Tel. (0 61 82) 7 78-0, Telex 4184055
Teletex (17) 6182915

Nordwürttemberg/Nordbaden
TEACH Hard- und Software Vertrieb GmbH + Co.
Siemensstraße 22, 7257 Ditzingen 1
Tel. (0 71 56) 30 01-0, Telex 7245248

Südwürttemberg/Südbaden
Binder Datentechnik GmbH
Geschäftsstelle VS-Villingen
Mönchweilerstraße 1, 7730 VS-Villingen
Tel. (0 77 21) 88-0, Telex 792568,
Fax (0 77 21) 88-359

Bayern
Binder Datentechnik GmbH
Geschäftsstelle München
Machtlfinger Straße 7, 8000 München 70
Tel. (089) 7 85 30 17, Telex 522300

Günther Sternberg

Drucker-Sharing mit geringem Aufwand

4 Drucker an 14 Rechnern mit dem Programm Easyprint

Was tut man, wenn man zwei oder mehrere PCs, die in nicht allzugroßer räumlicher Entfernung voneinander stehen, an Drucker oder Plotter anschließen soll? Mit Easyprint kann man diese Aufgabe lösen, ohne ständig die Ausgabegeräte an den jeweiligen aktiven PC umzustecken.

Mit Easyprint kann man an einen zentralen Rechner, im folgenden als File-Server bezeichnet, maximal vier Drucker oder Plotter anschließen. Das Besondere an Software Technologies Easyprint ist der völlige Verzicht auf spezielle Hardware-Komponenten. Für den Anschluß der Ausgabegeräte stehen dem Anwender die parallelen Schnittstellen LPT1 und LPT2, sowie die seriellen Schnittstellen COM1 und COM2 zur Verfügung. Auf diese vier Geräte können gleichzeitig über den File-Server maximal 14 Rechner (Satelliten) zugreifen. Dazu werden auf jedem Satelliten-Rechner maximal drei virtuelle, also nur scheinbar vorhandene, Ausgabegeräte definiert. Diese virtuellen Geräte sind vom Anwenderprogramm über die DOS-Schnittstellen LPT1, LPT2 oder LPT3 ansprechbar. Mehrere beim Booten des Rechners geladene residente Programme sorgen auf jedem Satelliten-Rechner für die Umleitung und Übertragung der Druckausgabe zum File-Server. Residente Programme im Speicher des File-Servers übernehmen die Zwischenspeicherung der empfangenen Daten in einem Spooler und die anschließende Ausgabe auf dem jeweiligen Gerät. Die Kommunikation zwischen File-Server und Satelliten-Rechner erfolgt über die RS232-Schnittstellen, z. B. über COM1 oder COM2.

Topologie des Netzes

Die Satelliten-Rechner schließt man sternförmig am File-Server an (*Bild 1*). Außerdem sind am File-Server die vier möglichen Ausgabegeräte angeschlossen, die von jedem Rechnern aus ansprechbar sein sollen. Selbstverständlich kann man an jeden Satelliten-Rechner noch zusätzliche Ausgabegeräte anschließen. Auf diese Geräte kann jedoch nur der betreffende Satelliten-Rechner zugreifen. Die Ausgabe einer Datei von einem Satelliten-Rechner über den File-Server auf das Gerät eines anderen Satelliten-Rechners ist nicht möglich.

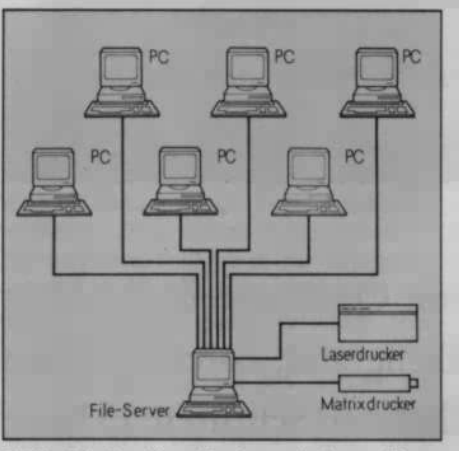


Bild 1: Die Satelliten-Rechner sind sternförmig am File-Server angeschlossen

Das Verbindungskabel zwischen File-Server und Satelliten-Rechner kann auf beiden Seiten entweder an die seriellen Schnittstellen COM1 bzw. COM2 oder an die seriellen Schnittstellen spezieller Zusatzkarten von Server Technology angeschlossen werden. Diese Zusatzkarten enthalten zwei oder sechs zusätzliche serielle Schnittstellen. Bei den Satelliten-Rechnern wird man sich aus Kostengründen meist der ohnehin vorhandenen seriellen Schnittstellen COM1 oder COM2 bedienen. Beim File-Server kommt man spätestens beim dritten Satelliten-Rechner um die Anschaffung einer Erweiterungskarte nicht mehr herum.

Funktionsweise von Easyprint

Als Verbindung dient ein einfaches serielles Schnittstellenkabel (*Bild 2...Bild 4*), dessen Länge laut Hersteller maximal 150 m betragen kann. Die Übertragung erfolgt mit 9600 Baud und einem Hardware-Handshake.

Auf jedem Satelliten-Rechner lassen sich ein bis drei virtuelle Drucker oder Plotter definieren. Jedem Gerät wird eine der drei DOS-Schnittstellen LPT1, LPT2 oder LPT3 zugeordnet. Eine Zuordnungstabelle stellt den logischen Zusammenhang zwischen dem virtuellen Gerät und dem realen Ausgabegerät am File-Server her. Eine Änderung der Zuordnung ist jederzeit vom Satelliten-Rechner aus möglich.

Sendet ein Anwenderprogramm nun eine Druckausgabe an eine DOS-Schnittstelle, die einem der virtuellen Geräte zugewiesen wurde, leiten die residenten Programme von Easyprint die Daten auf die serielle Schnittstelle um und übertragen sie blockweise zum File-Server. Die Easyprint-Routinen im File-Server speichern die eingehenden Daten in einer Datei. Der Satelliten-Rechner signalisiert dem File-Server das Ende der Druckausgabe, wenn seit der Ausgabe eines Zeichens durch das Anwenderprogramm eine vom Benutzer vorgegebene Wartezeit verstrichen ist. Der File-Server schließt dann die temporäre Datei und reiht sie im Spooler in die Liste der auszugebenden Dateien ein. Die Dateien des Spoolers werden nach dem FIFO-Prinzip (first in, first out) auf dem jeweiligen Gerät ausgegeben. Dabei kann Easyprint mehrere Ausgabegeräte gleichzeitig bedienen. Allerdings arbeitet dann ein auf dem File-Server im Vordergrund ablaufendes Anwenderprogramm deutlich langsamer.

Für Einsteiger

Zum Test lag das Einsteigerpaket von Easyprint für zwei Computer vor, das aus einem 110seitigen englischen Handbuch, zwei Disketten, einem ungefähr 7 m langen seriellen Schnittstellenkabel (zwei 25polige D-Buchsen) und zwei Adaptern (25polige D-Stecker und 9polige D-Buchse) bestand. Das Handbuch ist übersichtlich gestaltet und gut verständlich. Es beschreibt die

Funktionsweise von Easyprint, die Installation der Kabel, die Erstinstallation der Software für zunächst zwei Rechner und enthält Beispiele für eine Installation von zwei, drei und sieben PCs. Auf jedem Rechner müssen, außer den seriellen Schnittstellen zwei Diskettenlaufwerke oder ein Diskettenlaufwerk und eine Festplatte sowie DOS ab Version 2.0 vorhanden sein.

Wegen des hohen Speicherbedarfs der Programme von Easyprint (260 KByte beim File-Server und 240 KByte auf jedem Satelliten-Rechner) dürfte zumindest der File-Server normalerweise ohne Festplatte nicht auskommen, da auf dem File-Server noch zusätzlicher Speicherplatz für den Spooler benötigt wird. Die beim Booten über die Autoexec-Datei zu ladenden sechs residenten Programme belegen ungefähr 20 KByte des Hauptspeichers.

Einfache Installation

Doch nun zu den Erfahrungen bei der Installation selbst. Die Verbindung der beiden Rechner war eine Sache weniger Handgriffe. Zur Installation der Software muß für jeden Rechner eine eigene Diskette (jede Diskette muß eine andere Seriennummer enthalten) vorhanden sein. Diese Diskette ist ins Laufwerk A einzulegen und das Installationsprogramm aufzurufen. Alle nötigen Eingaben sind im Handbuch klar und detailliert beschrieben. Lediglich einige Zusatzabfragen (z. B. Diskette wirklich im Laufwerk A) sind im Handbuch nicht erwähnt.

Für die Installation muß man lediglich wissen, an welcher physikalischen Schnittstelle der Drucker und das serielle Verbindungskabel angeschlossen und unter welcher DOS-Schnittstelle der virtuelle Drukker am Satelliten-Rechner anzusprechen ist. Alles in allem ist die Installation einfach und relativ narrensicher. Nach dem Abschluß der Installation muß man alle Rechner booten, wobei sich das Netz automatisch aufbaut. Als ersten Schritt sollte man vom File-Server aus mit einem eigenen Testprogramm die Verbindung zu jedem Satelliten-Rechner überprüfen. Sind alle Verbindungen in Ordnung, kann die praktische Arbeit beginnen.

Erfahrungen im Betrieb

Zum Test wurden zwei Drucker (ein Nadeldrucker und ein Laserdrucker) an den File-Server angeschlossen und am Satelliten-Rechner lief eine Reihe von Programmen mit Druckausgaben ab. Alle Druckausgaben

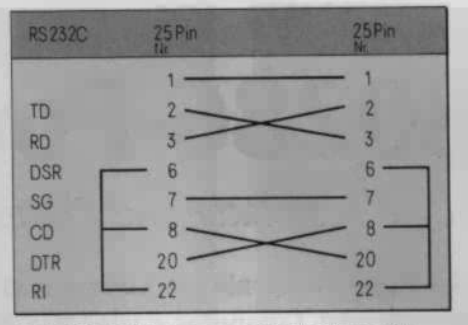


Bild 2: Null-Modem aus einer Leitung mit zwei 25poligen D-Sub-Buchsen

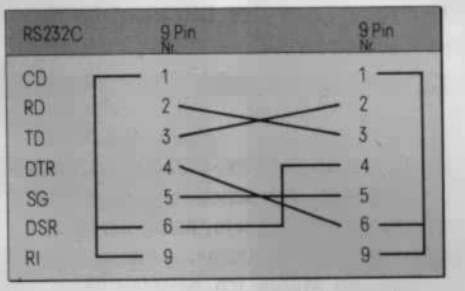


Bild 3: Null-Modem aus einer Leitung mit zwei 9poligen D-Sub-Buchsen

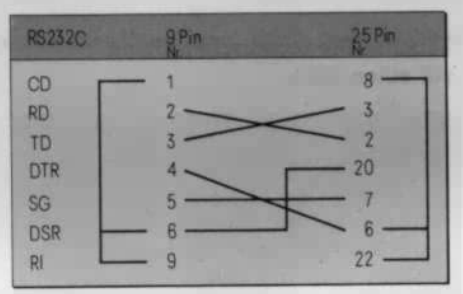


Bild 4: Null-Modem aus einer Leitung mit einer 25poligen und einer 9poligen D-Sub-Buchse

erschienen ohne Fehler auf dem richtigen Drucker. Auf dem File-Server, einem IBM-AT-kompatiblen Gerät mit 10 MHz Taktfrequenz und Festplatte, macht sich die im Hintergrund ablaufende Druckausgabe fast nicht bemerkbar. Wenn man jedoch gleichzeitig vom File-Server und vom Satelliten-Rechner aus drucken will, bekommt man ein Problem. Auf dem Bildschirm des File-Servers erscheint folgende Fehlermeldung:

```
Write fault error writing device PRN
Abort, Retry, Ignore?
```

Nun sitzt man in der Zwickmühle. Einerseits kann das Vordergrundprogramm auf dem File-Server nicht drucken, da das Gerät noch von Easyprint in Beschlag genommen wird, andererseits kann Easyprint wegen des Interrupt-24H-Fehlers nicht weiterdrucken. Damit Easyprint die Spoolerausgabe beenden kann, bleibt einem leider nichts anderes übrig, als die Druckausgabe des Anwenderprogramms auf dem File-Server abzubrechen.

Der Test förderte jedoch noch einen weiteren Fehler zutage. Alle Druckausgaben (Dateien) der Satelliten-Rechner müssen vor dem Ausschalten bzw. vor einem Neustart des File-Servers gedruckt werden. Nach einem Neustart des File-Servers sind zwar noch alle nicht gedruckten Dateien im Spooler vorhanden, doch unterbleibt die Ausgabe dieser Dateien. Easyprint hat diese Dateien nach einem Neustart schlicht und einfach vergessen.

Verwaltung von Druckformularen

Easyprint bietet noch eine Reihe wichtiger Funktionen, die für eine sinnvolle Nutzung eines Druckernetzwerks unentbehrlich sind. So kann man auf jedem Satelliten-Rechner für jeden virtuellen Drucker eine Initialisierungs- und eine Abschlußdatei vereinbaren. Zu Beginn jeder Druckausgabe wird zuerst die Initialisierungsdatei ausgegeben und am Ende die Abschlußdatei. Diese Dateien können zum Beispiel Steuerzeichen für den Drucker oder eine Titel- und eine Abschlußseite enthalten.

Ein weiteres Merkmal von Easyprint ist die Verwaltung von maximal vier Formularen. Ein Formular kann beispielsweise eine Seite mit einem aufgedruckten Briefkopf zur Verwendung auf einem bestimmten Drukker sein.

Alle vom Satelliten-Rechner empfangenen Druckausgaben für dieses Formular bleiben im Spooler des File-Servers, bis Easyprint mitgeteilt wird, daß sich das entsprechende Formular nun im Drucker befindet. Leider ließ sich diese Programmfunktion von Easyprint nicht testen, da das Programm zur Einstellung von Formularen immer der Meinung war, daß gar kein Drucker angeschlossen sei.

Drucker gut ausgelastet

Das Programmpaket, das in Deutschland von Wilke, Aachen, vertrieben wird, ist von seiner Leistung her, nicht zuletzt durch den Verzicht auf spezielle Hardware, für den Aufbau kleiner Druckernetzwerke (2...3 Satelliten-Rechner) geeignet.

Wer mit den Schwächen des Programms leben kann, kommt mit Easyprint zu einer preiswerten Lösung, um teuere Drucker besser auszulasten. □

Günther Sternberg

Tragbare 80286-Power

Laptop 300 SLC mit 20-MByte Festplatte und Plasmabildschirm

Insidern ist Rein-Elektronik als Lieferant elektronischer Bauteile und Computer-Peripherie bekannt. Mit dem Laptop 300 SLC gesellt sich Rein-Elektronik auch unter die PC-Anbieter. Beim Laptop 300 SLC sticht besonders der hochauflösende Plasmabildschirm hervor.

Bei dem nach den Vorstellungen von Rein-Elektronik in Japan konstruierten und gefertigten Gerät handelt es sich um einen Rechner der AT-Klasse, der mit einer mit 10 MHz getakteten 80286-CPU bestückt ist. Das Gerät enthält ein 3½-Zoll-Disketten-Laufwerk und eine 3½-Zoll-Festplatte mit 20 MByte Kapazität. Die weiteren technischen Daten sind in der *Tabelle* zusammengefaßt. Um es gleich vorweg zu sagen, wer einen tragbaren Computer für die U-Bahn, das Auto oder die grüne Wiese sucht, für den kommt der 300 SLC nicht in Frage. Der stromfressende Plasmabildschirm läßt einen netzunabhängigen Betrieb nicht zu. Aber gerade der Plasmabildschirm ist in meinen Augen auch zugleich das Besondere an diesem Gerät. Die Plasmaanzeigen sind in Punkto Kontrast und Lesbarkeit den Flüssigkristallanzeigen bei weitem überlegen. Selbst die neue Generation von Supertwist-LC-Displays – mit oder ohne Hintergrundbeleuchtung – ändert nichts an dieser Tatsache. Wer einem Anwender einmal bei der Arbeit an einem Laptop mit Flüssigkristallanzeige über die Schulter geschaut hat, der weiß wovon ich spreche. Bereits bei geringen Abweichungen vom optimalen Betrachtungswinkel erscheint die LC-Anzeige als eine einzige milchige Wand. Helles Tageslicht tut ein übriges, um LC-Anzeigen in einen „Rasierspiegel" zu verwandeln. Für viele Anwender (z. B. Handelsvertreter) ist es jedoch sehr wichtig, daß auch Umstehende einen guten Blick auf den Bildschirm haben. Die Plasmaanzeige des Rein-Laptops erfüllt diese Forderung bestens.

Doch sollte dieser Punkt für Sie nicht von so großer Bedeutung sein, so finden Sie vielleicht die Auflösung von 640 × 400 Punkten interessant. Damit sind im Textmodus 25 Zeilen mit je 80 Zeichen sauber und deutlich lesbar. Im Grafikmodus emuliert das Gerät EGA und CGA. Leider ist eine Darstellung der einzelnen Farben durch unterschiedliche Graustufen bei Plasmaanzeigen nicht möglich. Dadurch erscheint zum Beispiel eine gelbe Schrift auf blauen Hintergrund nur als heller Balken. Bei vielen Programmen, z. B. Open Access von SPI, lassen sich jedoch Vorder- und Hintergrundfarben von Menüs und Texten den eigenen Wünschen anpassen. Sollte dies nicht möglich sein, kann man fast immer den monochromen Anzeigemodus wählen. So mußte ich beispielsweise bei Turbo-Pascal (Version 3.01 und Version 4.0) den Schwarzweiß-Bildschirm mit 80 x 25 Zeichen auswählen.

Tabelle: Die technischen Daten auf einen Blick

CPU:
80286 mit 10 MHz Taktfrequenz
80287 als Option nachrüstbar

RAM-Speicher:
640 KByte standardmäßig
2 MByte mit Zusatzkarte nachrüstbar

Massenspeicher:
3½-Zoll-Laufwerk mit 1.2 MByte Kapazität
3½-Zoll-Laufwerk mit 20 MByte Kapazität
ein zweites Laufwerk mir 3½- oder 5¼-Zoll mit 360 KByte, 720 KByte oder 1,2 MByte ist extern anschließbar

Bildschirm:
Plasmabildschirm mit 80 × 25 Zeichen im Textmodus und 640 × 400 Punkten im Grafikmodus. Emulation von EGA und CGA.

Tastatur:
Deutsche Tastaturbelegung mit 82 Tasten

Schnittstellen:
Eine parallele Druckerschnittstelle (Centronics mit 25poliger D-Buchse), zwei serielle Schnittstellen (RS232 mit 9poligem D-Stecker)

Sonstiges
Platz für Zusatzkarte (100 × 100 mm). An den beiden Steckerleisten für die Zusatzkarte stehen alle Signale des AT-Busses zur Verfügung. Die Belegung entspricht allerdings nicht dem normalen AT-Bus.

Robustheit geht ins Gewicht

In *Bild 2* ist das Gerät im geschlossenen Zustand mit herausgezogenem Tragegriff und in *Bild 1* mit aufgeklapptem Bildschirm zu sehen. Das Gerät gehört zwar mit ungefähr 6,4 kg Gewicht zu den Schwergewichten der Laptops, doch läßt es sich dank seines stabilen und bequemen Tragegriffs mühelos tragen. Das hohe Gewicht kommt nicht zuletzt durch das sehr stabile anthrazitfarbene Spritzgußgehäuse zustande. Das 3½-Zoll-Laufwerk ist rechts vom Bildschirm, im Bodenteil des Geräts, eingebaut und gut zu erreichen. Unter dem Bildschirm befinden sich sieben Leuchtdioden mit folgender Bedeutung:

– Power	– Non ASCII
– Caps Lock	– HDD (Festplatte)
– Num Lock	– FDD (Diskette)
– Scroll Lock	

Die Leuchtdiode „Non ASCII" spiegelt den Zustand einer gleichnamigen Taste der Tastatur wieder. Mittels dieser Taste soll die Eingabe der Sonderzeichen erfolgen, die sonst nicht direkt erreichbar sind. Beim Testgerät stand der dazu nötige, speziell für dieses Gerät angepaßte, Tastaturtreiber jedoch noch nicht zur Verfügung. Die Tastatur selbst besteht aus 82 Tasten, die einen gut fühlbaren Druckpunkt aufweisen und einen stabilen, soliden Eindruck machen.
Beim Einschalten des Geräts fällt einem zuerst das unangenehme Geräusch des eingebauten Lüfters auf. Das ungleichmäßige Geräusch erinnert an das Pfeifen des Windes durch Ritzen. Die Ursache des Pfeifens liegt an der stömungstechnisch ungünstigen Form des Lüftergitters. Der Lüfter selbst ist leise. Bei Rein-Elektronik denkt man bereits über eine geeignetere Ausführung des Lüftergitters nach.

Das Gerät im Betrieb

Das Testgerät war bei mir drei Wochen lang fast täglich in Betrieb und es gab in diesem

Bild 1: Ein AT im handlichen Format mit hochauflösendem Bildschirm

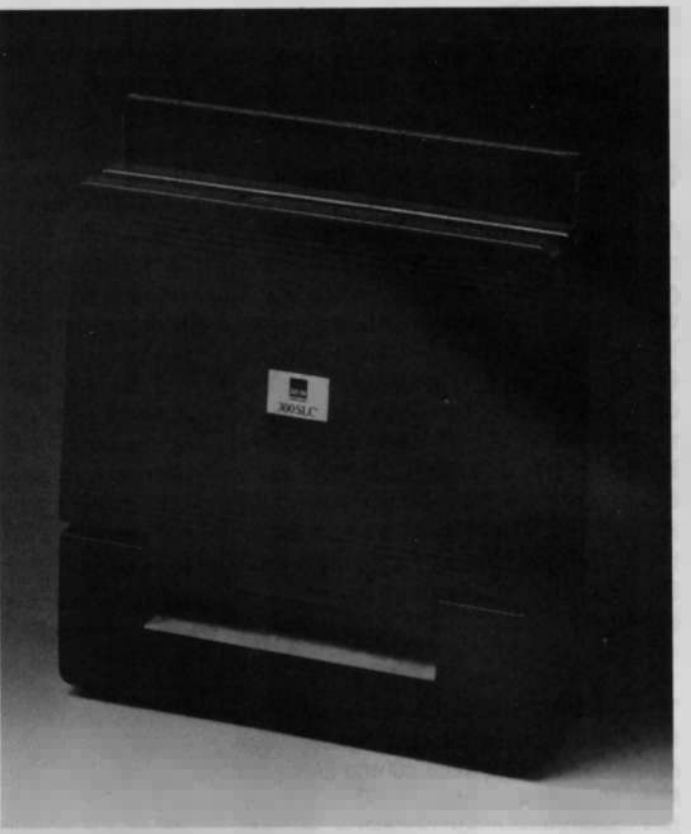


Bild 2: Das Gerät im geschlossenen Zustand (Maße: 320 × 320 × 94 mm³)

Zeitraum keinen Grund zu Beanstandungen. Während dieser Zeit wurde der Laptop mit verschiedenen Programmen getestet, wobei keinerlei Schwierigkeiten oder Kompatibilitätsprobleme auftraten. Bei jedem Wechsel auf meinen eigenen AT mit 6 MHz Taktfrequenz kam mir dieser richtig langsam vor. *Bild 3* zeigt einen Vergleich des 300 SLC mit einigen anderen getesteten Geräten.

Eine angenehme Überraschung gleich in der ersten Stunde bildete das 3½-Zoll-Diskettenlaufwerk, daß auf 1,2 MByte ausgelegt ist. Abgesehen von der mit dem Gerät mitgelieferten 3½-Zoll-Betriebssystem-Diskette hatte ich nur Software auf 5¼-Zoll-Disketten. Zum Umkopieren hatte ich an einem meiner Rechner lediglich ein 3½-Zoll-Laufwerk mit 360 KByte oder 720 KByte Kapazität. Der Laptop erkannte jedoch selbstständig jedes der drei möglichen Formate.

Schnittstellen und Zusatzkarten

Das Gerät ist standardmäßig mit zwei seriellen Schnittstellen (RS232 mit 9pol. AT-Stecker) und einer parallelen Druckerschnittstelle (Centronics mit 25pol. D-Buchse) ausgerüstet. Eine weitere Buchse ermöglicht den Anschluß eines externen 3½- oder 5¼-Zoll-Diskettenlaufwerks. Dieses Laufwerk muß allerdings über eine eigene Stromversorgung verfügen. Als einzigen Kritikpunkt muß man die leichte Verwechselbarkeit dieses Anschlusses mit der parallelen Schnittstelle erwähnen. Beide besitzen eine 25polige D-Buchse und liegen zudem direkt nebeneinander. Alle Anschlüsse befinden sich gut erreichbar auf der Rückseite des Geräts und sind ausreichend beschriftet.

Das Gerät besitzt zwar keine Steckplätze für Einsteckkarten im üblichen Format, jedoch kann man eine kleine Zusatzkarte (Größe ungefähr 100 x 100 mm) einbauen. Diese Zusatzkarte wird von der Rückseite des Geräts aus eingeführt und man kann an einem Abschlußblech Steckverbinder nach außen anbringen. An zwei Steckerleisten stehen der Zusatzkarte alle Signale des AT-Busses zur Verfügung. Auf einer solchen Zusatzkarte ließe sich beispielsweise eine

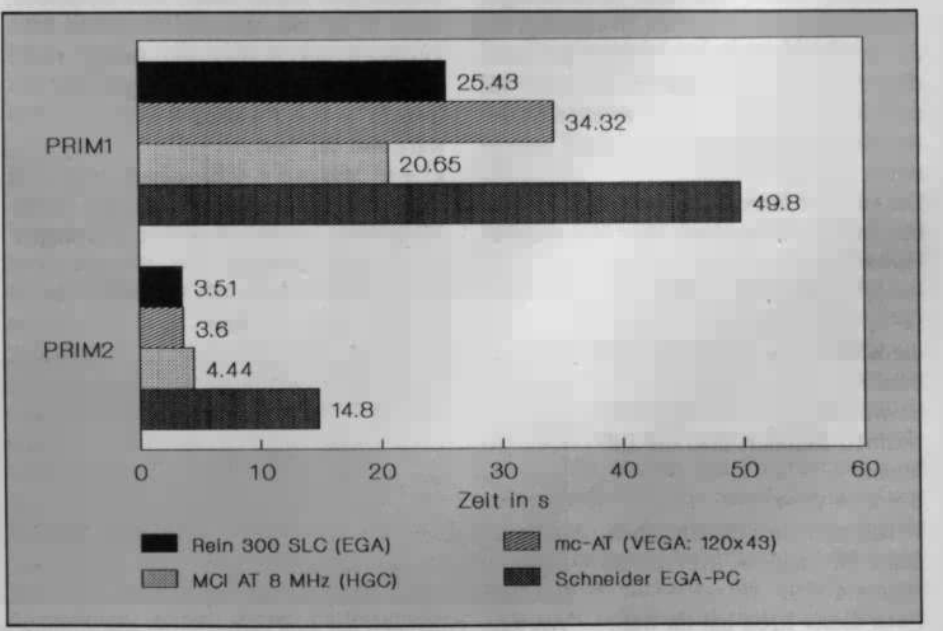


Bild 3: Der 300 SLC braucht den Vergleich mit anderen Systemen nicht zu scheuen

IEEE-Schnittstelle realisieren. Von Rein-Elektronik selbst wird eine Zusatzkarte mit einer 2-MByte-Speichererweiterung angeboten.

Bedienungshandbuch in Deutsch

Mit dem Gerät wird ein deutsches Bedienungshandbuch mit ungefähr 110 Seiten geliefert. Dieses Handbuch enthält alle wichtigen und nötigen Informationen zur Handhabung des Geräts. So findet man Erklärungen zum Umgang mit den Disketten, den eingebauten Laufwerken, dem externen Laufwerk, der Tastatur, dem Setup-Programm und Hinweise zur Installation des Betriebssystems (einschließlich RAM-Floppy). Positiv hervorzugehen sind die Tabellen mit den Anschlußbelegungen aller Steckerleisten. Auch die genaue Belegung der Steckerleisten für die Zusatzkarte, sowie die genauen Maße der Karte gehen aus diesen Tabellen hervor.

Das Handbuch zum DOS-Betriebssystem und zum GW-Basic-Interpreter waren zum Zeitpunkt des Tests noch nicht verfügbar. Ein beiliegendes Schreiben versprach eine rasche Nachlieferung eines deutschen Handbuchs.

Die mitgelieferte Diskette enthält neben dem DOS-Betriebssystem (Version 3.21 in Englisch) ein Setup-Programm, GW-Basic (Version 3.22) und eine Utility zum Parken der Festplatte. Das Setup-Programm enthält allerdings noch einen kleinen Fehler. Wird nur das Datum oder die Uhrzeit verändert, so schreibt das Programm die neuen Daten nicht in den batteriegepufferten Setup-Speicher der Echtzeituhr. Erst wenn man pro forma einen der weiteren Systemparameter verändert, erfolgt die Abspeicherung der neuen Daten.

Plus und Minus

Den gleichen Computer gibt es unter der Bezeichnung Rein 300 SLC/40 mit einer 40-MByte-Festplatte (NEC D3146H702), die eine Zugriffszeit von etwas unter 40 ms hat.

Insgesamt kann man den Laptop 300 SLC als einen hochwertigen Rechner der AT-Klasse bezeichnen, dessen Leistungsfähigkeit weit über den Standard hinausgeht. Abgesehen von den Kritikpunkten (pfeifender Lüfter und Fehler im Setup-Programm) macht das Gerät einen sehr ordentlichen und soliden Eindruck. Der einzige Wermutstropfen ist der noch relativ hohe Preis (etwas über 9000 DM bei dem Modell mit 20-MByte-Festplatte). Die Möglichkeit einen AT mit EGA-Bildschirm unter dem Arm spazieren zu tragen, hat derzeit eben noch seinen Preis. □

Auch dieser Kreis hat Ecken

In mc 12/85, Seite 84 wurde ein Algorithmus zum Kreiszeichnen vorgestellt, der denen in mc 4/85, Seite 60 und mc 8/85, Seite 60 überlegen ist.

Der erste Oktant mit $0° <= \varphi <= 45°$ wurde bei dem besten Algorithmus wie folgt gezeichnet:

– Mache immer einen Schritt in die positive y-Richtung

– Wenn Du dabei aus dem Kreis herausgerätst, so mache einen Schritt in die negative x-Richtung.

– Der Oktant ist fertig gezeichnet, wenn x und y (ohne Beachtung der Mittelpunktskoordinaten) den gleichen Wert erreicht haben.

Um den Oktanten mit $45° < \varphi <= 90°$ zu zeichnen, sollen die drei Anweisungen leicht abgewandelt werden. Wenn man die Symmetrie des Kreises bezüglich der Geraden durch den Mittelpunkt mit $\varphi = 45°$ ausnutzt, so erhält man den zweiten Oktanten einfach durch Vertauschen der berechneten x- und y-Werte.

Das in Basic formulierte Programm vermeidet durch Anwendung der binomischen Formeln

$(x+1)^2 = x^2 + 2x + 1$ und

$(x-1)^2 = x^2 - 2x + 1$

die wiederholte Quadrierung in dem Test $x^2 + y^2 > r^2$?

Es wird aber am Anfang r^2 berechnet. Definiert man die Differenz

$d = x^2 + y^2 - r^2$,

so läßt sich auch das umgehen, denn für (x,y) = (r,0) ist zu Beginn d = 0.

Doch dieser Kreis hat im wahrsten Sinne des Wortes immer noch seine Ecken, und zwar in den Punkten $(x,y) = (\pm r,0)$ und $(x,y) = (0,\pm r)$. Für y = 0 ergibt sich x = r, aber für y = +1 ist schon x = r–1. Genauer gesagt: für

$1 <= y <= [SQR(2r-1)]$

gilt

x = r–1,

denn $x = SQR(r^2-y^2)$ wird stets abgerundet. Dabei steht [z] für die größte ganze Zahl, die nicht größer als z ist, und SQR für die Wurzel. Die Ecken lassen sich vermeiden, wenn in der zweiten Anweisung erst beim Verlassen des Kreises mit Radius r+0.5 nach links gegangen wird. Wegen

$(r+0.5)^2 = r^2 + r + 0.25$

wäre also der Test zu

$x^2 + y^2 > r^2 + r + 0.25$

abzuwandeln. Betrachten man den Grenzfall genauer, so stellt man fest, daß wegen

$x <= r$ der Grenzfall

$y^2 + (x-1)^2 = r^2 + r - 2x + 1 >$

$r^2 - r + 0.25 = (r-0.5)^2$

impliziert. Das heißt, auch der Punkt (x–1,y) ist von der idealen Kreislinie mit dem Radius r weniger als 0.5 entfernt. Er liegt sogar näher am idealen Kreis, so daß man getrost den Test abwandeln darf zu

$x^2 + y^2 >= r^2 + r$?

Damit erhält man für das Zeichnen eines Kreises das abgedruckte Programm.

Schon der Kreis mit dem Radius 4 zeigt, daß der Fall $x^2 + y^2 = r^2 + r$ wirklich auftreten kann. Man nehme (x,y) = (4,2); dieser Punkt hat vom Kreis mit r=4 den Abstand 0.472, während der Punkt (3,2) nur den Abstand 0.394 hat. Das hat zur Folge, daß beim Zeichnen konzentrischer Kreise mit Radien $0 <= r <= r_{max}$ der Punkt (4,2) nie gezeichnet wird.

Für r_{max} gegen unendlich beträgt der Anteil der Rasterpunkte, die auf keinem Kreis liegen,

$1 - SQR(8)/\pi \approx 0.0997$

Das sind knapp 10 %. Überlegen kann man sich das folgendermaßen: Für festes y gibt es im ersten Oktanten bei $r \approx SQR(2)y$ den ersten gesetzten Rasterpunkt und für $r = r_{max}$ den letzten.

Also liegen für festes y ungefähr $r_{max} - SQR(2)y$ Rasterpunkte auf den gezeichneten Computerkreisen. Summation über y ergibt

$r_{max}^2 / SQR(8)$,

während im Achtelkreis mit dem Radius r_{max} ungefähr

$\pi r_{max}^2 / 8$

Rasterpunkte liegen.

Der Befehl PLOT im Listing muß dem jeweiligen Basic-Dialekt angepaßt werden.

Dr H. Englisch, Dr. H.-J. Herrler

```
10 INPUT "mittelpunkt xm, ym = " : XM, YM
20 INPUT "radius          r = "; R
50 D=-R : X=R : Y=0
57 PLOT XM+X, YM+Y : PLOT XM-X, YM+Y
58 PLOT XM+X, YM-Y : PLOT XM-X, YM-Y
59 PLOT XM+Y, YM+X : PLOT XM-Y, YM+X
60 PLOT XM+Y, YM-X : PLOT XM-Y, YM-X : IF X<=Y THEN END
61 D=D+2*Y+1 : Y=Y+1 : IF D>=0 THEN D=D-2*X+1 : X=X-1
62 GOTO 57
```

Das Basic-Programm mit dem verbesserten Algorithmus zum Kreiszeichnen

Lore Schönrock

WordPerfect 4.2

oder die Komplexität der Textverarbeitung

Der professionelle Einsatz einer Textsoftware muß von der komfortablen Bedienung der benötigten Verarbeitungsfunktionen abhängig gemacht werden. Mit der kompetenten Betreuung durch den Händler läßt sich dann die Lernphase bis zur qualifizierten Nutzung des Programmes verringern.

Wer sich der Illusion hingibt, daß er sich nur an das PC-System mit der installierten Textverarbeitung zu setzen braucht und nach wenigen Tastenproben den Leistungsumfang der Software abstecken und nutzen kann, unterschätzt die Komplexität der Aufgabe. Das ist eine Anforderung, die wohl keine der auf dem Software-Markt gehandelten professionellen Textverarbeitung erfüllen kann. Um das hohe Gestaltungs- und Verarbeitungsniveau zu gewähren, müssen oft komplexe Strukturen bis zur Auslösung der Funktion durchlaufen werden. Denn das Geheimnis für das Verständnis der Funktionsabläufe liegt in der Programmlogik, die begriffen sein will, damit die daraus resultierende beste Lösungsvariante für eine Aufgabenstellung entwickelt werden kann. Aus dieser Überlegung heraus kann es nur eine logische Schlußfolgerung geben: den baldmöglichsten optimalen Einsatz der Software kann nur eine Schulung vorbereiten.

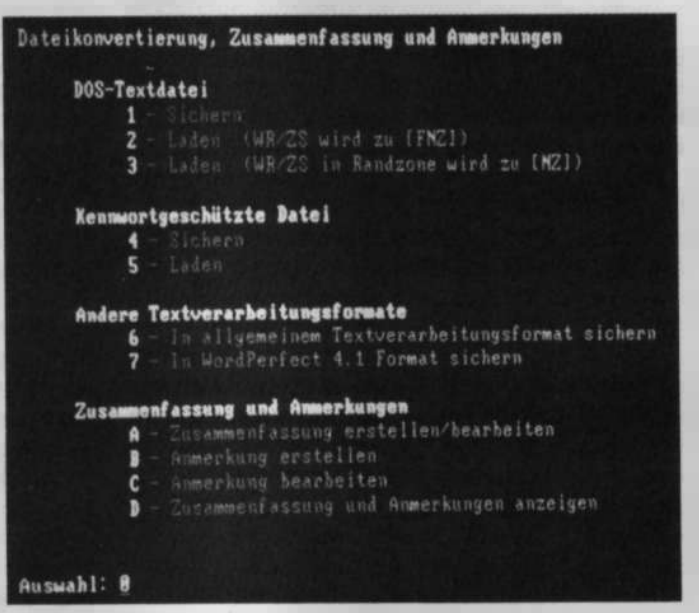


Bild 1. Das erweiterte Dateimenü

Die Probleme bei der Einführung einer neuen Software bis hin zu ihrer optimalen Nutzung vom Anwender ist den Software-Entwicklern und Herstellern nicht fremd. Viele schulen aus diesem Grund ihre Vertriebspartner oder arbeiten mit Schulungspartnern zusammen, die sich nach den Seminarunterlagen des Herstellers richten sollen.

Einige gehen sogar noch einen Schritt weiter. Die Firma WordPerfect beispielsweise, deren gleichnamiges Programm als professionelle Textsoftware in Deutschland verbreitet ist, legt besonderen Wert darauf, das geeignete Umfeld für die Integration ihrer Programme zu schaffen. Alle Programme, ob Tabellenkalkulation, Terminplanung und -Verwaltung oder auch Datenbank, haben aus diesem Grund die gleiche Benutzeroberfläche. Entsprechend ist die Einführung weiterer Produkte überall dort begünstigt, wo bereits mit einem Programm aus dem Hause WordPerfect gearbeitet wird. Denn die Grundbegriffe sind vorauszusetzen und die Lernphase kann dadurch nochmals stark verkürzt werden.

Der professionelle Einsatz

WordPerfect zählt zu den Textprogrammen, die ein sehr umfangreiches Funktionsspektrum für den professionellen Einsatz bieten. Es ist ausgeschlossen hier eine vollständige Beschreibung vornehmen zu wollen, und schon gar nicht im Detail. Interessant zu sein, scheint mir hingegen, einige ausgewählte Problemstellungen anzureißen, die zur Lösung bestimmter Aufgaben beitragen, wie das Schreiben, Gestalten und Archivieren wissenschaftlicher Abhandlungen mit WordPerfect.

Eine Fülle von Grundfunktionen der Textbearbeitung bringt diese professionelle Software mit. Erwähnen möchte ich nur solche wie Halbzeilenschaltung, Korrekturkennung, Blättern von einer Dokumentenseite zur nächsten, denn das sind Funktionen, die das Arbeiten mit langen Texten sehr vereinfachen. Doch für die Gestaltung wissenschaftlicher Arbeiten müssen weitaus mehr vorhanden sein. Funktionen der Textverarbeitung sind eine wichtige Voraussetzung für das zu schreibende Inhaltsverzeichnis, für die oft zu ändernde Absatz- und Seitennumerierung, für die fortlaufend zu numerierenden Anmerkungen und Verweise, für die Gestaltung der Kopf- und Fußzeilen, für die Aufnahme in Quellenverzeichnisse, Indizes u. ä. Zusätzlich wird die Arbeit durch eine Rechtschreibprüfung und ein erweiterbares Wörterbuch erleichtert. Nicht selten gehören Tabellen oder Grafiken, mathematische oder auch chemische Formeln dazu, deren Gestaltung gleichermaßen vorgenommen sein möchte. Fast alle dieser aneinandergereihten Funktionen sind mit WordPerfect 4.2 verfügbar.

Doch lassen Sie uns schrittweise vorgehen. Das Programm ist auf einer Mindestkonfiguration von einem IBM-PC, AT oder kompatiblen Rechner mit 256 KByte RAM lauffähig. Für die umfangreichen Funktionen, wie z. B. Rechtschreibprüfung oder Stichwortverzeichnis, die viel Speicherplatz benötigen, sollte von einer erweiterten Hardware-Konfiguration mit Festplatte ausgegangen werden. Erst damit wird letztlich ein komfortables Arbeiten mit dem Textprogramm gewährleistet.

Alle Funktionen der Software sind über die Funktionstasten und in einer Kombination mit der CTRL-, ALT- sowie SHIFT-Taste abrufbar. Mit Starten des Programmes wird man aufgefordert, die Schablone mit Kurzhinweisen für die Funktionstasten auf die Tastatur zu legen. Außerdem hilft die mehrseitige Referenzkarte, die ebenfalls dem Textpaket beigelegt ist, besonders bei weniger häufig gebrauchten Funktionen.

Muß eine bestimmte Textstelle korrigiert werden, will man sich nicht lange durch den Text bis dahin tasten. In WordPerfect gibt es deshalb die Funktion „Blättern". Bis zur Korrekturstelle kann man sich seitenweise vortasten. Für eine nächste Überarbeitung kann die eingefügte Textstelle mit einer Korrekturkennung versehen werden. Dazu markiert man die Textpassage als Block und ruft die Korrekturkennung als Befehl auf, die sich hinter ALT-F5 verbirgt. Am linken Rand wird ein vertikaler Strich auf die Einfügung hinweisen. Ähnlich kann auch ein zu streichender Text „durchgestrichen" werden. Dafür ist die gleichnamige Funktion zuständig, die den Textteil mit Gedankenstrichen überdruckt. Ist der Text endgültig überarbeitet, lassen sich alle auf diese Weise gesetzten Markierungen mit ALT-F5, 6 löschen.

Professionelle Textsoftware schließt meist auch ein Wörterbuch ein. Bei WordPerfect umfaßt es ca. 120 000 Wörter und ist erweiterbar. Bei einer halbautomatischen Rechtschreibprüfung wird der Text mit den Wörtern im Wörterbuch verglichen. So können einerseits Fehler leichter aufgespürt, und andererseits im gleichen Arbeitsgang neue Wörter ins Lexikon eingegeben werden. Nach und nach wird man den Wortschatz vervollständigt haben.

Gewöhnlich ist es für Textprogramme ein Problem, mathematische oder chemische Formeln zu schreiben. Auch WordPerfect kann keine Diagonalen und Kreise zeichnen. Strukturformeln lassen sich also nicht vollständig gestalten. Aber Tabellen können umrahmt und Organigramme gezeichnet werden. Außerdem gibt es halbe Zeilenschaltungen, die behilflich sein können. Damit kann ein leeres Zeilen- und Zeichenraster erstellt werden, in dem sich der Cursor zum Schreiben der mathematischen Formel oder Gleichung frei nach oben und unten bewegen läßt. (Ist kein Raster aus Leerzeichen angelegt, bleibt der Cursor am Anfang der Zeile stehen.) Es empfiehlt sich, die Formel oder Gleichung in einer Musterdatei vorzufertigen und sie dann an die richtige Position in den Text zu setzen. Hat man die vier Ecken des Rasters mit einem Zeichen markiert, können später die nicht verwendeten Leerzeilen gelöscht werden, wenn die Formel in den Text eingebunden wird. Das Zeilenformat muß nach der Formel für den sich anschließenden Text wieder geändert werden. Während der Bildschirm die Formel einzeilig dargestellt, wird sie korrekt ausgedruckt. Rechen- und Sonderzeichen stehen zur Verfügung. An dieser Stelle sei auch erwähnt, daß einfache Rechnungen mit WordPerfect vorgenommen werden können. Jedes beliebige Zeichen des ASCII-Zeichensatzes, das auf der Tastatur nicht vorgesehen ist, läßt sich auf einer Buchstabentaste definieren und ist mit einem Tastendruck darstellbar.

Ein Verzeichnis muß sein

Ob vor, während oder nach Fertigstellung des Manuskriptes die Überschriften über bestimmte Textabschnitte abgefaßt und aufgeschrieben werden, ist für deren Übernahme in das Inhaltsverzeichnis völlig unerheblich. Erst dann, wenn man sich mit den Verzeichnissen befassen möchte, ist es gerade recht. Die Überschriften müssen für das Inhaltsverzeichnis unter ALT-F5 markiert werden. Damit das Inhaltsverzeichnis bereits optisch in Kapitel, Abschnitte und Unterabschnitte untergliedert wird, öffnet sich eine Menüleiste, auf der einzugeben ist, zu welcher Stufe diese Überschrift gehören soll. Es sind fünf Abstufungen verfügbar, die Einrückungen verursachen. Diese Einrückungen werden auf Tabulatorstopps gesetzt. Die standardmäßig vorgegebenen Positionen müssen nicht verwendet, sondern können frei bestimmt werden. Lange Überschriften nehmen entsprechend mehrere Zeilen im Inhaltsverzeichnis ein. Die Seitenangabe wird rechtsbündig neben dem jeweiligen Titel erfolgen. Je nach Vorstellung des Autors kann das Inhaltsverzeichnis am Anfang des Textes oder am Ende stehen. An der definierten Stelle wird auf Tastendruck das Verzeichnis eingefügt.

Zu jeder Monographie wird meist am Ende ein Literatur- oder Quellenverzeichnis benötigt. Hier werden Hinweise über die Zitate gegeben, die in Gruppen von maximal 16 Sachgebieten zusammengefaßt und darin alphabetisch geordnet werden können. Demzufolge sollten sich alle Literaturhinweise zum Sachgebiet „Grundlagen" auch in dem entsprechend benannten Abschnitt befinden und in alphabetischer Ordnung aufgeführt sein. Da es nicht üblich ist, eine häufig zitierte Quelle ebenso häufig im Verzeichnis aufzuführen, kann sie mit WordPerfect unter einer Kurzform markiert werden, so daß alle dazugehörenden Seitenangaben rechts daneben stehen werden.

Im Text selbst sind diese Hinweise in den Fußnoten festgehalten, die jeweils am Ende der Seite ausgedruckt werden. Es gibt auch Endnoten, in denen diese Informationen festgeschrieben stehen, aber wie der Name schon sagt, werden diese dann am Ende des Textes gedruckt. Bei der normalen Textdarstellung sind Fuß- und Endnoten nicht am Bildschirm angezeigt. Für ihre Gestaltung hält WordPerfect eine Reihe Funktionen bereit. So läßt sich beispielsweise der Zeilenabstand innerhalb und zwischen den Noten bestimmen. Die Numerierung erfolgt automatisch aufsteigend.

Verzeichnisse für Abbildungen, Grafiken u. ä. können ebenfalls erstellt werden. Dazu dient in der Menüleiste unter ALT-F5 der Befehl „Liste". Es lassen sich bis zu fünf verschiedene Listen dieser Art gestalten, doch kann ein Begriff immer nur für eine davon markiert werden.

Analog dazu wird auch das Stichwortverzeichnis erstellt. Hierbei ist es erlaubt, Haupt- und Untertitel zu bestimmen. Ein Haupttitel könnte z. B. Computer-Zeitschrift heißen und ein dazugehörender Untertitel mc. Hinter dem eingerückten mc würde dann die Seitenangabe folgen. Nun ist das Markieren von Wörtern für einen Index recht aufwendig, vor allem dann, wenn die Wörter mehrmals auftreten. Um Doppelungen zu vermeiden, könnte man zwar mit der Funktion „Suchen" arbeiten, aber damit hätte man das Wort noch nicht im Index aufgenommen.

Eine einfache Variante wäre, alle Begriffe zunächst einmal zu markieren. Diese Wörter sollten dann in einer Konkordanzdatei erfaßt werden. Beim Erstellen des Stichwortverzeichnisses wird im Text nach den Wörtern und Begriffen gesucht, die in dieser Datei aufgenommen sind, und automatisch übernommen. Dabei wird von vorneherein gewährleistet, daß jedes Wort nur einmal erscheint und Doppelungen ausgeschlossen sind. Die Konkordanzdatei ist eine Textdatei, in der jede Wendung auf einer Zeile stehen muß und mit einer festen Zeilenschaltung abgeschlossen wird. Den Wörtern und Begriffen darin können, wie im normalen Text auch, Indexeinträge zugeordnet werden. Beim Generieren des Verzeichnisses werden sie automatisch in den Text übernommen. Für das Stichwortverzeichnis bei identischen Begriffen sind die Angaben der Konkordanzdatei bindend. Um das Erstellen des Indexverzeichnisses noch mehr zu beschleunigen, sollten die Begriffe in der Konkordanzdatei alphabetisch sortiert werden. Der „Sortier"-Befehl verbirgt sich hinter CTRL-F9.

Ein Macro muß her

Es gibt auch eine andere Lösung für diese Aufgabe. Ein Macro müßte erstellt werden, das nach dem vorgegebenen Wort sucht und es nach Auffinden für den Index markiert, d. h. mit einem Indexeintrag versieht. Unter Macro versteht man, die Zusammenfassung mehrerer Funktionen oder Befehle in einer Datei. Mit Aufrufen der Datei werden die darin stehenden Befehle abgearbeitet. So sind die Macros ein wichtiges Hilfsmittel zur schnelleren Lösung be-

stimmter Aufgaben. Man müßte also in dem obigen Beispiel ein Wort, das ins Verzeichnis aufgenommen werden soll, eintippen und kann alles weitere dem Macro überlassen. Es findet das Wort und versieht es mit der nötigen Markierung.
Wenn man möchte, könnte man die Funktion „Macro" sogar wohlwollend als Kommandosprache bezeichnen, denn sie automatisiert und vereinfacht für den Anwender ständig wiederkehrende Bedienungsabläufe.
Da mehrere Macros sich auch noch zusammenfassen lassen, kann damit ein komplexer Arbeitsgang in Auftrag gegeben werden. Das Arbeiten mit diesem Hilfsmittel ist nur dem Anwender zu empfehlen, der WordPerfect schon sehr gut kennt. Eine gut angelegte Dokumentation über jeden einzelnen Tastenanschlag in umfangreicheren Macros ist dringend erforderlich, besonders, da sie mit WordPerfect nach ihrer Fertigstellung weder einzusehen noch zu ändern sind. Das einzige, was sich auch später noch ändern läßt, ist der Name des Macros. (Nur mit dem Macro-Editor, der Bestandteil des Software-Paketes WordPerfect Library ist, lassen sich inhaltliche Korrekturen vornehmen.)
Auf diese Art und Weise können Formate, Seitenparameter u.ä. einmal erarbeitet, festgelegt und immer abrufbar gehalten werden. In einer solchen Datei würden alle Handgriffe stehen, wie Randeinstellung (linker und rechter Rand), Tabulatorstopps, Trennvorschlag „Ein", Zeilenabstand, Blocksatz, Schritteilung.

Theoretisch könnte man auch so komplexe Operationen wie das Sortieren von Adressen für den Serienbrief mit Macroverkettungen abarbeiten lassen. Das widerspricht jedoch der Programmlogik. Denn mit der steigenden Komplexität der Macrokette sinkt die Verarbeitungsgeschwindigkeit. Macros sind deshalb für eine schnelle Abwicklung dieser Aufgabe nicht geeignet. Denn dafür gibt es in WordPerfect den Mischvorgang. Dabei werden aus der Sekundärdatei alle ausgewählten Adressen nacheinander mit dem Brief in der Primärdatei gemischt und ausgedruckt. Man sollte sich davor hüten, alles mit Macros automatisieren zu wollen. Will man trotzdem den Mischvorgang umgehen, kann man die Adressen mit dBase bearbeiten und sie für den Serienbrief aus der Datenbank übernehmen. So führen viele Wege zum gleichen Ziel. Mit der nötigen Programmkenntnis und dem Verständnis für dessen Logik wird sich der Anwender für den effizienten Weg entscheiden und die Programmvorzüge nutzen.

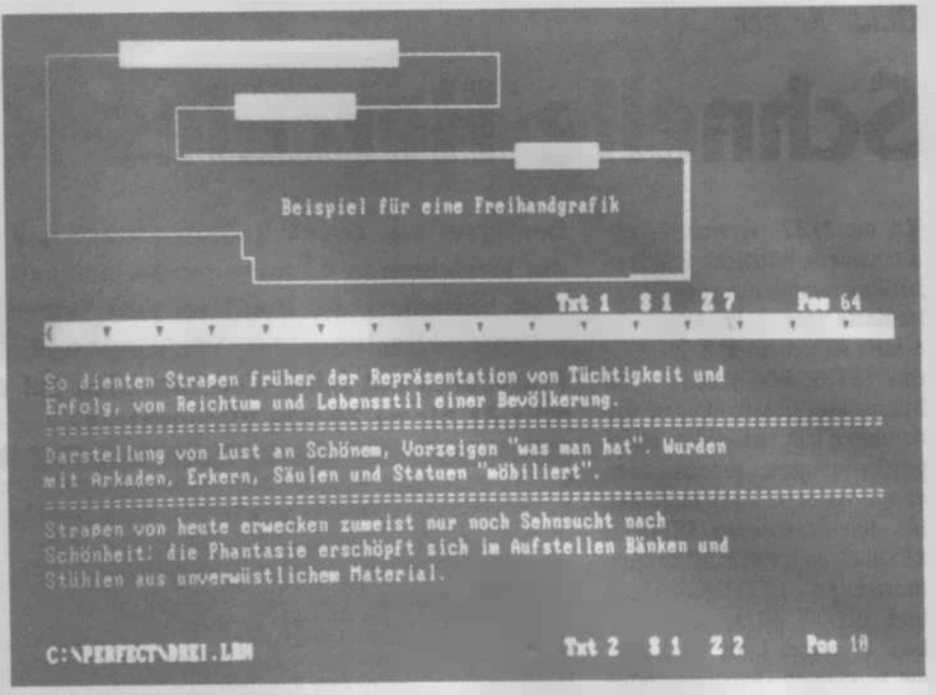


Bild 2.
Mit zahlreichen Linienarten lassen sich über Tastendrücke einfache Grafiken erstellen

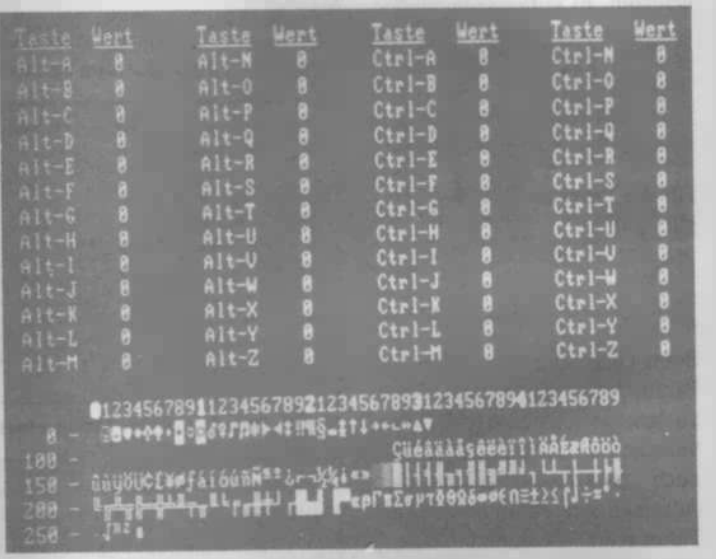


Bild 3.
Jede Taste läßt sich mit einem Zeichen des ASCII-Vorrates belegen

Die Nummer stimmt

Vor dem Ausdruck des umfangreichen Textes muß an die Seiten- und Absatznumerierung sowie eine mitgeführte Kopfzeile gedacht werden. In der Kopf- oder Fußzeile möchte man die Bezeichnung des Kapitels mitführen und gewiß auch die Seitennummer. WordPerfect ist auf mehrere Textzeilen in den Kopf-, aber auch in den Fußzeilen eingestellt. Der Text kann mit Fettdruck, Unterstreichungen und mit einer anderen Schrift versehen werden. Weitaus interessanter ist jedoch, daß es zwei verschiedene Kopf- und auch zwei verschiedene Fußtexte sein dürfen. Gedacht ist dabei an die Variante für gerade und ungerade Seiten, die abwechselnd unterschiedliche Inhalte ausweisen sollen. Beim Arbeiten am Text schreibt Word-Perfect eine Sicherungskopie auf Diskette. An eines muß jedoch der Anwender noch selbst denken: Lange Texte sollten zweckmäßiger in mehrere Dateien geschrieben und gespeichert werden. Einige Gründe sprechen dafür, wie höhere Geschwindigkeit beim Bearbeiten, bestimmte Textstellen lassen sich schneller finden, Teilaufgaben lassen sich leichter delegieren und nicht zuletzt sei auch die Datensicherheit erwähnt.
Bleibt anzufügen, daß WordPerfect als eines unter wenigen Programmen Postscript unterstützt. Das fertige Manuskript könnte also von einem Laserdrucker ausgegeben werden und als Druckvorlage für die Vervielfältigung dienen. □

Guido Krüger

Schnelle Hercules-Routinen

Der Wert von Grafik-Routinen steht und fällt mit der erreichbaren Ausgabegeschwindigkeit. Kritisch sind besonders die Routinen zum Setzen und Löschen eines Bildpunktes. Die vorgestellten Pascal-Prozeduren mit Inline-Code schaffen auf einem 8-MHz-AT bis zu 20 000 Bildpunkte/s.

In mc 2/87 [1] wurden verschiedene nützliche Routinen zur Programmierung der Hercules-Karte vorgestellt. Auf einem 8-MHz-AT schafft z. B. die Routine SETRESDOT() etwa 6400 Bildpunkte/s. Mit einigen Änderungen läßt sich die Zeichengeschwindigkeit jedoch noch enorm steigern. Dazu muß man bei den Prozeduren SETDOT, RESDOT und SWITCHDOT die Prozeduren BYTEOFFSET(x,y) und BITOFFSET(x) umschreiben. In erster Linie muß man die langsamen Divisionsbefehle durch schnelle Schiebebefehle ersetzen. Es gilt:

```
X div 8 = X shr 3
Y div 4 = Y shr 2
Y mod 4 = Y and 3
$2000 * R = R shl 0dh
```

Für die Prozedur BITOFFSET(x) gilt:

```
X mod 8 = X and 7
8 - S = $80 shr S
```

Beim Originalprogramm ist die Parameterübergabe zum Stack zu langsam, da zu viele Daten verschoben werden. Dies gilt auch für die Übergabe der Adressen von Parametern (Var-Parameter). Schneller ist es, Werte-Parameter zu übergeben. Außerdem funktioniert die Übergabe von Adreß-Referenzen nicht bei allen Kontrollstrukturen von Turbo-Pascal.
Mit der Deklaration „absolute" bindet man die Variablen an die Offset-Adressen 0105/0107 des Code-Segments. Da hier das Borland-Logo steht, kann man es gefahrlos überschreiben. Die Methode funktioniert sowohl bei Speicher-Compilierung als auch bei COM-Dateien.
Erstaunlicherweise brachte auch das Einfügen des Codes in den Quellcode der Turbo-Pascal-Prozedur mit der Inline-Anweisung noch eine weitere Geschwindigkeitssteigerung gegenüber dem External-Befehl. Man kann die Geschwindigkeit leicht auf ca. 22 500 Bildpunkte/s steigern, wenn man die Compiler-Überprüfungen mit folgender Steueranweisung ausschaltet:

```
(*$C-,R-,V-,U-,A+,X+,K-*).
```

Eine weitere Steigerung der Geschwindigkeit auf ca. 24 000 Bildpunkte/s erreicht man durch globale, aber absolute Deklaration der Variablen x und y und parameterlosen Aufruf der Zeichenfunktionen, was jedoch in Verbindung mit der vorherigen Beschleunigung nicht immer funktioniert.
Der reine Inline-Code braucht zur Ausführung 102 Taktzyklen. Bei einem 8-MHz-AT entspricht das 12,75 µs. Um die Portabilität zu gewährleisten, wird in den Prozeduren *(Bilder 1...5)* nur 8088/8086-Code verwendet.

```
procedure setdot(x,y:integer);
var
   a:integer absolute dseg:$0105;
   b:integer absolute dseg:$0107;
begin
     a:=x;b:=y;
     inline($b8/$07/$01/$8b/$f0/$8b/$14/$b8/$05/$01/
            $8b/$f0/$8b/$1c/$8b/$c2/$81/$e2/$03/$00/
            $b1/$0d/$d3/$e2/$d1/$eb/$d1/$eb/$d1/$eb/
            $03/$da/$d1/$e8/$d1/$e8/$b1/$5a/$f6/$e1/
            $03/$d8/$8b/$0c/$81/$e1/$07/$00/$b8/$80/
            $00/$d3/$e8/$1e/$b9/$00/$b8/$8e/$d9/$8b/
            $f3/$8a/$1c/$0a/$d8/$88/$1c/$1f);
end;
```

Bild 1. Eine Prozedur zum Setzen eines Bildpunktes

```
procedure resdot(x,y:integer);
var
   a:integer absolute dseg:$0105;
   b:integer absolute dseg:$0107;
begin
     a:=x;b:=y;
     inline($b8/$07/$01/$8b/$f0/$8b/$14/$b8/$05/$01/
            $8b/$f0/$8b/$1c/$8b/$c2/$81/$e2/$03/$00/
            $b1/$0d/$d3/$e2/$d1/$eb/$d1/$eb/$d1/$eb/
            $03/$da/$d1/$e8/$d1/$e8/$b1/$5a/$f6/$e1/
            $03/$d8/$8b/$0c/$81/$e1/$07/$00/$b8/$80/
            $00/$d3/$e8/$1e/$b9/$00/$b8/$8e/$d9/$8b/
            $f3/$8a/$1c/$35/$ff/$ff/$22/$d8/$88/$1c/$1f);
end;
```

Bild 2. Diese Prozedur löscht einen Bildpunkt

```
procedure setdot(x,y:integer);
var
   a:integer absolute dseg:$0105;
   b:integer absolute dseg:$0107;
begin
     a:=x;b:=y;
     inline($b8/$07/$01/$8b/$f0/$8b/$14/$b8/$05/$01/
            $8b/$f0/$8b/$1c/$8b/$c2/$81/$e2/$03/$00/
            $b1/$0d/$d3/$e2/$d1/$eb/$d1/$eb/$d1/$eb/
            $03/$da/$d1/$e8/$d1/$e8/$b1/$5a/$f6/$e1/
            $03/$d8/$8b/$0c/$81/$e1/$07/$00/$b8/$80/
            $00/$d3/$e8/$1e/$b9/$00/$b8/$8e/$d9/$8b/
            $f3/$8a/$1c/$32/$d8/$88/$1c/$1f);
end;
```

Bild 3. Ein Bildpunkt wird invertiert

Literatur

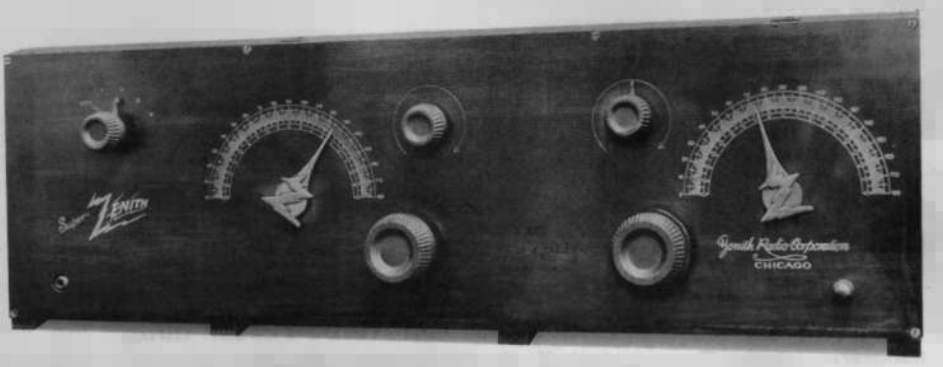
[1] *Bremer, Rudolf:* Turbo-Pascal bedient Hercules-Karte. mc 1987, Heft 2, Seite 46.

```
procedure resbyte(x,y:integer);
var
   a:integer absolute dseg:$0105;
   b:integer absolute dseg:$0107;
begin
     a:=x;b:=y;
     inline($b8/$07/$01/$8b/$f0/$8b/$14/$b8/$05/$01/
            $8b/$f0/$8b/$1c/$8b/$c2/$81/$e2/$03/$00/
            $b1/$0d/$d3/$e2/$d1/$eb/$d1/$eb/$d1/$eb/
            $03/$da/$d1/$e8/$d1/$e8/$b1/$5a/$f6/$e1/
            $03/$d8/$1e/$b9/$00/$b8/$8e/$d9/$8b/
            $f3/$b0/$00/$88/$04/$1f);
end;
```

Bild 4. Eine Prozedur setzt ein Byte

```
procedure setbyte(x,y:integer);
var
   a:integer absolute dseg:$0105;
   b:integer absolute dseg:$0107;
begin
     a:=x;b:=y;
     inline($b8/$07/$01/$8b/$f0/$8b/$14/$b8/$05/$01/
            $8b/$f0/$8b/$1c/$8b/$c2/$81/$e2/$03/$00/
            $b1/$0d/$d3/$e2/$d1/$eb/$d1/$eb/$d1/$eb/
            $03/$da/$d1/$e8/$d1/$e8/$b1/$5a/$f6/$e1/
            $03/$d8/$1e/$b9/$00/$b8/$8e/$d9/$8b/
            $f3/$b0/$ff/$88/$04/$1f);
end;
```

Bild 5. Ein Byte wird gelöscht



Super Zenith, erstes Radio von 1918

So sah vor 70 Jahren das erste Zenith-Gerät aus, nachdem 1918 Ralph Matthews und Karl Hassel in Chicago die Zenith Radio Corporation gegründet hatten. Es war ein Radioempfänger, der für damalige Verhältnisse so leistungsfähig war, daß die Produktion bis 1924 auf ein Gerät pro Tag gesteigert werden konnte.
Seitdem hat Zenith seine Rolle als Hersteller leistungsfähiger elektronischer Geräte nicht wieder aufgegeben. Immer wieder ist es den Zenith-Ingenieuren gelungen, bahnbrechende Erfindungen zu entwickeln, die modernste Technologie für den Anwender nicht nur nutzbar, sondern auch erschwinglich machte.

1926 Zenith erfindet das erste Radio, das am Wechselstrom-Netz betrieben werden kann.

1927 Zenith erfindet die Stations-Taste, mit der ein Sender auf Knopfdruck eingestellt werden kann.

1934 Zenith erfindet die Einknopf-Abstimmung, mittels derer durch Drehen eines einzigen Knopfes die Sender gewählt werden und wie sie bis heute üblich ist.

1940 Zenith nimmt die erste kommerzielle UKW-Station der Welt in Betrieb und ermöglicht damit den Durchbruch der Frequenzmodulation als Übertragungsverfahren.

1941 Zenith baut das erste Kofferradio, das tragbar und batteriebetrieben ist. Dieses Gerät wurde als wichtigstes Kommunikationsmittel zur Polarexpedition von Donald B. MacMillan gewählt.

1948 Zenith stellt dem Markt das erste Fernsehgerät vor: ein Schwarz-weiß-Gerät mit einer runden Bildröhre.

1950 Zenith erfindet die Infrarot-Fernbedienung für Fernsehgeräte.

1960 Zenith ist größter amerikanischer Hersteller von Fernsehgeräten. Jeder dritte amerikanische Fernseher ist ein Zenith.

1971 Zenith erfindet ein völlig neues Verfahren, um die Farbdarstellung an Farbbildröhren (Beamer) erheblich zu verbessern.

1979 Zenith gründet den Geschäftsbereich Zenith data systems und stellt seinen ersten 8-bit Microcomputer vor, den Z-89.

1981 Zenith entwickelt den Z-110, einen 8- und 16-bit Computer, der zu den damaligen Standards höchste Kompatibilität bietet.

1983 Zenith ist der erste namhafte Hersteller der Welt, der einen vollkompatiblen PC herstellt, den Z-151 und den Portable Z-161.

1985 Zenith stellt seinen Z-248 dem Markt vor, den ersten 80286-PC mit Null-Wartezyklen-Technologie.

1986 Zenith erfindet das Supertwisted-LCD, wie es bis heute in den Portables Z-180 eingebaut ist und bisher in Größe und Darstellungsqualität noch von keinem anderen Hersteller gezeigt wird.

1987 Zenith ist erster Hersteller der Welt, der einen vollkompatiblen PC mit 80386 Prozessor und echtem 32-bit Bussystem baut. Eine Leistung, die ebenfalls bis heute unerreicht blieb.

1988 Zenith beginnt die Massenproduktion des FTM-Monitors, eines Farbbildschirmes, der vorne völlig flach ist und aufgrund neuer Technologien bisherige Farbmonitore hinsichtlich Kontrast und Helligkeit um fast das Doppelte überbietet.

Auch jetzt nach 70 Jahren ist es unser Ziel, neueste Forschungsergebnisse der Elektronik in bezahlbaren Anwendernutzen umzusetzen. Über 37000 Mitarbeiter in 30 Produktionsstätten rund um die Welt arbeiten an diesem Ziel. Denn Erfolg ist schließlich eine Geschichte, die nicht von ungefähr kommt.
Wenn Sie wissen wollen, welchen Nutzen Ihnen die technologisch führenden PC's von Zenith bringen, dann schreiben Sie uns oder rufen Sie an. Wir senden Ihnen gern einen Übersichtsprospekt und nennen Ihnen die Adresse eines Zenith-Fachhändlers ganz in Ihrer Nähe, der Sie über Einsatzmöglichkeit und Anwendungen gern ausführlich berät.

Zenith Z-386 mit FTM-Monitor, der leistungsstärkste unter den Zenith PC's

ZENITH | data systems

Die 100% Computer

Zenith data systems GmbH, Robert-Bosch-Str. 32-38
6072 Dreieich-Sprendlingen
Telefon: 0 61 03/39 05-0, Telefax: 0 61 03/3 19 31, Telex: 417986

Ingo Eickmann, Bernd Ewert

Parameterübergabe bei Turbo-Pascal

Manchmal ist es selbst bei Turbo-Pascal nötig einen Programmteil als Assembler-Routine zu realisieren. Anhand einfacher Beispiele wird ein komplexes Kapitel der höheren Programmierkunst transparent. Ein Bonbon für Praktiker ist das Statistik-Programm mit 8087-Befehlen.

Die Übergabe von Parametern (Werteparameter) bzw. der Adresse von Parametern (Referenzparameter) vom Hauptprogramm an ein Unterprogramm und die Rückgabe von Funktionsergebnissen vom Unterprogramm an das Hauptprogramm läuft bei Turbo-Pascal immer über den Stack. Um eine einwandfreie Kommunikation zwischen Hauptprogramm und Unterprogramm zu gewährleisten, muß bekannt sein, in welchem Format die Parameter auf dem Stack abgespeichert werden.

Bei den Prozessoren der 8086-Familie „wächst" der Stack nach unten, d.h. von den höheren Speicheradressen zu den niedrigeren.

Bei dem Aufruf eines Unterprogramms durch die Befehle Procedure () bzw. Function () werden auf dem Stack die zu übergebenden Daten oder die Adresse des „Datenblocks" und die Rücksprungadresse abgelegt. Ein Unterprogramm sollte (gemäß dem Turbo-Handbuch) zuerst das Basisseiten-Register (BP) sichern und es anschließend mit dem Wert des Stackpointers (SP) laden. Dies geschieht durch die Befehle:

```
PUSH BP
MOV BP, SP
```

Durch den PUSH-Befehl wird das Basisseiten-Register auf dem Stack abgelegt und der Stackpointer um zwei erniedrigt, so daß der erste Parameter vier Bytes vom SP entfernt liegt. Wenn dem Unterprogramm die Variablen A und B beim Funktions- bzw. Prozeduraufruf übergeben werden, sieht der Stack wie in *Bild 1* aus.

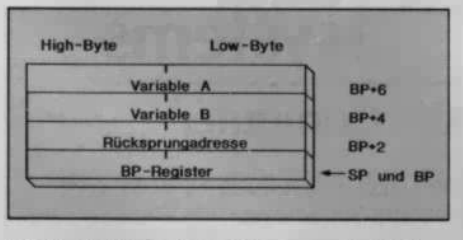


Bild 1. Ein Turbo-Pascal-Programm hat bei einem Prozeduraufruf zwei Variablen auf den Stack gelegt. Der Stackpointer zeigt auf das im Unterprogramm gerettete BP-Register

Adressierung der Parameter

Der Stackpointer bildet zusammen mit dem Stacksegment (SS) eine absolute Adresse. Dies ermöglicht den direkten Zugriff auf die im Stack liegenden Parameter, welche durch den Unterprogrammaufruf übergeben werden. Da PUSH- und POP-Befehle den Wert des Stackpointers ständig verändern, ist eine Adressierung der Parameter über den Stackpointer nicht sinnvoll. Daher wird beim Eintritt in die externe Assemblerroutine das BP-Register auf die gleiche Adresse wie der Stackpointer gesetzt. Nun kann man immer über einen konstanten Offset auf die Parameter zugreifen. So bewirkt zum Beispiel der Befehl MOV AX,[BP+4] das Laden des AX-Registers mit dem im Stack befindlichen Wert der Integer-Zahl B.

Die Anzahl an Bytes, die ein Parameter auf dem Stack belegt, hängt vom dessen Typ ab. Eine Aufstellung über Speicherbedarf und Datenstruktur der möglichen Parameter

Tabelle: Verzeichnis der Variablentypen unter Turbo-Pascal

Variablentyp	Größe der übergebenen Parameter auf dem Stack
INTEGER	2 Byte
BYTE	2 Byte (höherwertiges Byte = 00)
REAL	6 Byte bzw. 8 Byte (Turbo87)
BOOLEAN	2 Byte (LSB=0 false, LSB=1 true)
CHAR	2 Byte
STRING	erstes Byte=Stringlänge anschl. Zeichen des Strings
MENGEN	32 Byte
ZEIGER	4 Byte (Segment, Offset der Variablen) NIL=0000,0000
ARRAY	Segment und Offset des 1. Bytes
RECORD	Segment und Offset des 1. Bytes
VAR	Segment und Offset des Referenzparameters

```
PROCEDURE Test1(A,B:INTEGER); EXTERNAL 'T1.BIN';

        PUSH BP
        MOV  BP,SP
        MOV  BX,[BP+4]    ;B in BX laden
        MOV  AX,[BP+6]    ;A in AX laden

   z.B. ADD AX,BX         ;Operationen, die vom
                          ;Unterprogramm ausge-
                          ;führt werden sollen.

        MOV  SP,BP        ;Alten Zustand wiederherstellen
        POP  BP
        RET  0004         ;Rücksprung zu Turbo und
                          ;Stack freimachen.
```

Bild 2. Eine Turbo-Pascal-Prozedur ruft ein Assembler-Unterprogramm auf. Der Stack ist wie in Bild 1 aufgebaut

```
FUNCTION Test2 (A:INTEGER):INTEGER; EXTERNAL 'T2.BIN';

            PUSH BP
            MOV  BP,SP
            MOV  BX,[BP+4]     ;A in BX einlesen
                 :
                 :
            MOV  AX,ERGEBNIS   ;Das Ergebnis
            MOV  SP,BP         ;wird in AX an Turbo
            POP  BP            ;zurückgegeben.
            RET  0004          ;Stack freigeben
```

Bild 3. Diese Funktion liefert als Ergebnis eine Integer-Zahl. Die Assembler-Routine sichert mit dem ersten Befehl das BP-Register

```
FUNCTION Test3:BOOLEAN; EXTERNAL 'TST3.BIN';

            PUSH BP
            MOV  BP,SP

      z.B.  XOR  AX,AX         ;Z-Flag setzen

            MOV  SP,BP
            POP  BP
            RET  0001
```

Bild 5. Eine Turbo-Pascal-Funktion erwartet ein Ergebnis vom Typ Boolean

```
PROCEDURE TEST3 (VAR A:INTEGER); EXTERNAL 'T3.BIN'

            PUSH BP
            MOV  BP,SP
            LES  SI,[BP+4]
            MOV  AX,ES:[SI]    ;A in AX einlesen
                 .
                 .
            MOV  ES:[SI],AX    ;zurückspeichern des
                               ;Ergebnisses in A.
            MOV  SP,BP
            POP  BP
            RET 0004
```

Bild 7. Eine Turbo-Pascal-Prozedur übergibt einen Referenzparameter

ben. Dazu gibt man beim RET-Befehl die Anzahl der vom Stack zu entfernenden Bytes an. Im Beispiel belegen zwei Integer-Variablen vier Bytes, der entsprechende Befehl muß also RET 0004 lauten.

Weitere Beispiele

Die Behandlung einer Funktion unterscheidet sich von der einer Prozedur in einem wesentlichen Punkt. Turbo-Pascal reserviert bereits beim Funktionsaufruf einen zusätzlichen Speicherbereich auf dem Stack für das Funktionsergebnis (*Bild 3*). Aus dem *Bild 4* geht hervor, daß Turbo-Pascal ab

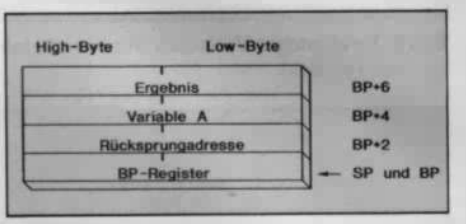


Bild 4. So sieht der Stack nach den ersten beiden Befehlen der Assembler-Routine

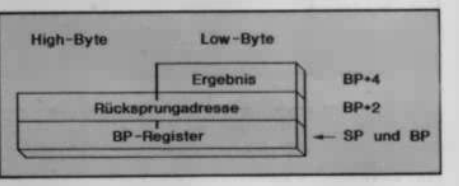


Bild 6. Auf dem Stack ist ein Byte für das Funktionsergebnis reserviert

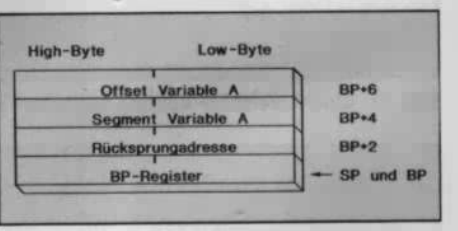


Bild 8. Auf dem Stack liegt die Adresse der Variablen A im Segment-Offset-Format

von Turbo-Pascal (MS-DOS, Version 3.0) finden Sie in der *Tabelle*. Die Reihenfolge, in der Turbo-Pascal die Parameter auf den Stack legt, ist vom Kopf der jeweiligen Prozedur bzw. Funktion abhängig. Die im Kopf zuerst deklarierten Parameter werden auch zuerst auf dem Stack abgelegt und befinden sich dementsprechend an den höchsten Adressen des Stacks. Da der Stack von den höheren Adressen zu den niedrigeren Adressen „wächst", sind die zuerst gespeicherten Parameter beim Aufruf des Unterprogramms am weitesten von dem Byte entfernt, auf das der Stackpointer bzw. das BP-Register zeigt.

In *Bild 2* sieht man den Kopf einer Prozedur, die zwei Werte-Parameter vom Typ Integer an ein Unterprogramm übergibt. Das in Bild 2 schematisch angedeutete Assembler-Unterprogramm adressiert dementsprechend die beiden übergebenen Variablen A und B mit BP+6 und BP+4.

Vor dem Rücksprung aus dem Unterprogramm müssen der Stackpointer und das BP-Register wieder auf ihre alten Werte zurückgesetzt werden. Dies geschieht mit den Befehlen MOV SP, BP und POP BP. Der Stackpointer zeigt jetzt wieder auf die Rückkehradresse.

Beim Rücksprung zum Hauptprogramm muß man den für die Variablenübergabe benutzten Stackbereich wieder freigegeben. Dazu gibt man beim RET-Befehl die Anzahl der vom Stack zu entfernenden Bytes an. Im Beispiel belegen zwei Integer-Variablen vier Bytes, der entsprechende Befehl muß also RET 0004 lauten.

Weitere Beispiele

Die Behandlung einer Funktion unterscheidet sich von der einer Prozedur in einem wesentlichen Punkt. Turbo-Pascal reserviert bereits beim Funktionsaufruf einen zusätzlichen Speicherbereich auf dem Stack für das Funktionsergebnis (*Bild 3*). Aus dem *Bild 4* geht hervor, daß Turbo-Pascal ab SP+6 bzw. BP+6 zwei Byte für das Funktionsergebnis reserviert. Dieser Bereich wird jedoch nur bei einigen Variablentypen wirklich benötigt. So wird eine Integer-Zahl als Funktionsergebnis nicht über den Stack, sondern über das AX-Register an das Hauptprogramm übergeben. Beim RET-Befehl des Programms in Bild 3 sind deshalb die zwei zusätzlich reservierten Bytes zu berücksichtigen.

Ähnlich verhält es sich bei einer Funktion mit dem Ergebnistyp Boolean (*Bild 5*). Hier erfolgt die Rückgabe des Funktionswerts über das Zero-Flag. Der Wert „True" wird durch das gesetzte und der Wert „False" durch das gelöschte Zero-Flag dargestellt.

```
FUNCTION TEST5 (a,b :INTEGER):REAL; EXTERNAL 'T5.BIN';

        PUSH    BP
        MOV     BP,SP

        FINIT
        FILD    WORD PTR [BP+6]         ;A in R7  laden
                                        ;    ==> ST = 7
        FILD    WORD PTR [BP+4]         ;B in R6  laden
                                        ;    ==> ST = 6
        LEA     DI,[BP+8]               ;STACK Adresse
        PUSH    SS                      ;für Ergebnis
        POP     ES                      ;in ES:DI
        FDIVP   ST(1),ST                ;R7 = R7 / R6
                                        ;   ==> ST = 7
        FSTP    QWORD PTR ES:[DI]       ;R7 ==> ERGEBNIS

        MOV     SP,BP                   ;Rücksprungzustand
        POP     BP                      ;herstellen und
                                        ;zurück zum
        RET     0004H                   ;Hauptprogramm.
```

Bild 9. Zwei Integer-Variablen werden an eine Assembler-Routine übergeben. Das Ergebnis ist vom Typ Real.

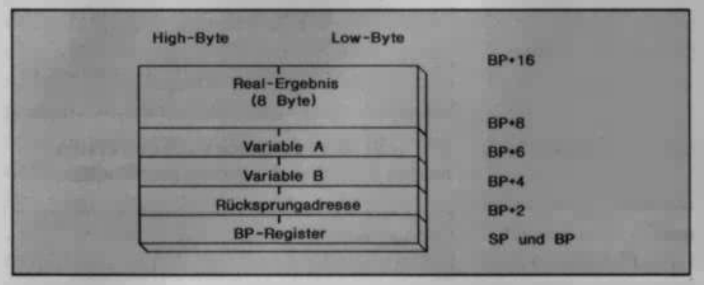


Bild 10. Das Ergebnis vom Typ Real belegt acht Byte auf dem Stack

```
PUSH BP
MOV  BP,SP
SUB  SP,8               ;Reservierung von 8 Byte auf
  :                     ;Stack
MOV  [BP-2],AX          ;Zwischenspeichern von AX und
MOV  [BP-4],BX          ;BX im reservierten Bereich
  :
MOV  SP,BP
POP  BP
RET  xxxx
```

Bild 11. Eine Assembler-Routine reserviert auf dem Stack acht Byte für lokale Variablen

```
PROGRAM STATISTIK;

CONST FELDGROESSE = 20;

TYPE  FELDTYP = array [1..FELDGROESSE] of real;
VAR   DATEN : FELDTYP;
      I     : INTEGER;
      MW,ST : REAL;

{---Externes Unterprogramm zur Berechnung von Mittelwert---}
{---------------   und Standardabweichung -----------------}

PROCEDURE Statistik; EXTERNAL 'STATISTIK.BIN';

FUNCTION  Mittelwert (VAR WERTEFELD : FELDTYP ; N : INTEGER):
real;EXTERNAL Statistik[0];

FUNCTION  Standardabw (VAR WERTEFELD:FELDTYP ; MW:REAL;
N:INTEGER):REAL;EXTERNAL Statistik[3];

BEGIN
FOR i:=1 TO FELDGROESSE DO
  Daten[i] := 4*i;
  MW := Mittelwert(Daten,FELDGROESSE);
  writeln('Mittelwert           : ',MW:8:4);
  ST:=STANDARDABW(Daten,MW,FELDGROESSE);
  writeln ('Standardabweichnung  : ',ST:8:4);
END.
```

Bild 12. Das Statistik-Programm ruft ein Assembler-Unterprogramm auf

Da Turbo-Pascal vor dem Aufruf des Unterprogramms durch Dekrementieren des Stackpointers ein Byte auf dem Stack für das Funktionsergebnis bereitstellt (*Bild 6*), muß durch den Befehl RET 0001 der Ursprungszustand wieder hergestellt werden.
Referenzparameter sind eine weitere Methode, um den Wert von Variablen aus dem Unterprogramm an das Hauptprogramm zurückzugeben. Dazu muß man im Turbo-Pascal-Programm der Variablen in der Parameterliste das Wort VAR vorausstellen. Es wird dann nicht der Wert der Variable, sondern deren absolute Adresse im Hauptspeicher an die Subroutine übergeben. Das Beispiel eines entsprechenden Unterprogramms ist *Bild 7* zu entnehmen. In *Bild 8* ist der Aufbau des Stacks dargestellt. Mit dem Befehl LES reg,[BP+n] (Load Pointer Using ES) wird das Segment der Variablen ins ES-Register und deren Offset in das Register reg geladen. Die Variable selbst läßt sich dann durch die indirekte Adressierung über ES:[reg] bearbeiten.
Während Funktionsergebnisse vom Typ Byte, Integer oder Booelan direkt, also unter Umgehung des Stacks, an das Hauptprogramm zurückgegeben werden, ist das bei anderen Typen nicht mehr möglich. Hier kommt nun der für das Funktionsergebnis im Stack reservierte Bereich zum tragen. In diesen Bereich muß der zu übertragende Wert vor Beendigung des Unterprogramms abgelegt werden. Der Stack darf also nur noch bis zum Ergebnis freigegeben werden, damit der Stackpointer nach dem Rücksprung auf das erste Byte des Ergebnisses zeigt. Ein Beispiel für die Übergabe einer Variablen vom Typ Real vom Assembler-Unterprogramm an das Turbo-Pascal-Programm ist in *Bild 9* und *Bild 10* zu sehen. Real-Variablen sind in Turbo-Pascal in der Version ohne Unterstützung des Numerikprozessors (8087) sechs Bytes und mit 8087-Unterstützung acht Bytes lang. Im Beispiel wird ein numerischer Coprozessor angesprochen, also Real-Zahlen mit acht Byte Länge verwendet.

Lokale Variablen auf dem Stack

Wichtig beim Schreiben von externen Unterprogrammen ist, daß hier keine Adressierung verwendet wird, die sich auf eine absolute Speicheradresse bezieht, da der Code des Unterprogramms an einer für den Programmierer unbekannten Stelle im compilierten Programm eingebunden wird. Es gibt im wesentlichen zwei Methoden, um dennoch auf lokale Variablen zugreifen zu können. Bei der ersten Methode läßt man einfach einen bestimmten Speicherbereich im Unterprogramm für eigene Variablen

```
_PROG segment byte
      assume CS:_PROG
      ORG 100H

START: JMP MITTELWERT     ; Einsprungoffset =0
       JMP STANDARDABW    ; Einsprungoffset =3

MITTELWERT PROC NEAR
        PUSH BP
        MOV  BP,SP
        MOV  CX,WORD PTR [BP+4]    ;Feldgröße in CX
        FINIT
        FLDZ
        CMP  CX,0
        JZ   FERTIG
        LES  SI,[BP+6]             ;Adresse des ersten
                                   ;Feldelementes
ADDIERE:FADD QWORD PTR ES:[SI]     ;Element laden
        ADD  SI,8                  ;Zeiger auf nächstes Element
        LOOP ADDIERE               ;Werte aufaddieren
        FIDIV WORD PTR [BP+4]      ;durch Anzahl dividieren
FERTIG: LEA  DI,[BP+10]            ;Ergebnis auf Stack schreiben
        PUSH SS
        POP  ES
        FST  QWORD PTR ES:[DI]
        MOV  SP,BP
        POP  BP
        RET  0006                  ;Zurück zum Hauptprogramm
MITTELWERT ENDP

STANDARDABW PROC NEAR
        PUSH BP
        MOV  BP,SP
        MOV  CX,WORD PTR [BP+4]    ;Feldgröße in CX
        FINIT
        FLDZ                       ;Null laden
        FLD  QWORD PTR [BP+6]      ;Mittelwert
        CMP  CX,0
        JZ   READY
        LES  SI,[BP+14]            ;Adresse des ersten
                                   ;Feldelementes
                                   ;Summe : (x(n)-Mittelwert)^2
ADD2:   FLD  QWORD PTR ES:[SI]     ;Element laden
        FSUB ST,ST(1)              ;X-MW
        FMUL ST,ST(0)              ;quadrieren
        FADDP ST(2),ST             ;aufaddieren
        ADD  SI,8                  ;Zeiger auf nächstes Element
        LOOP  ADD2

READY:  FINCSTP
        FIDIV WORD PTR [BP+4]      ;Summe durch Anzahl dividieren
        FSQRT                      ;Wurzel ziehen
        LEA   DI,[BP+18]           ;Ergebnis auf Stack schreiben
        PUSH  SS
        POP   ES
        FST   QWORD PTR ES:[DI]
        MOV   SP,BP
        POP   BP
        RET   0014                 ;Zurück zum Hauptprogramm

STANDARDABW ENDP

 _PROG ENDS
  end  start
```

Bild 13. Die Assembler-Routine spricht den 8087 an

frei und ermittelt den Offset des Speicherbereichs erst während des Programmlaufs. Der Quellcode sieht wie folgt aus:

```
L
      CALL HILF
      (1. Variable)
      (2. Variable)
      .............
HILF  Label NEAR
      POP DI
L
```

Das DI-Register enthält nun die Adresse der ersten Variable.

Die zweite Methode hat den Vorteil, daß im ohnehin begrenzten Code-Segment kein Speicherplatz für Variablen belegt werden muß. Der benötigte Platz wird auf dem Stack reserviert, indem man den Stackpointer einfach um die verlangte Anzahl an Byte dekrementiert (*Bild 11*). Die erste lokale Variable befindet sich dann im Stack an der Stelle BP-2. Bei dieser Methode ist bereits im voraus der Offset aller lokalen Variablen bekannt.

Mehrfacheinsprünge

Eine als „External" deklarierte Datei kann gleichzeitig mehrere Unterprogramme enthalten. Diese können, wie das Beispielprogramm (*Bild 12* und *Bild 13*) zur Berechnung von zwei einfachen statistischen Funktionen zeigt, durch Angabe eines Einsprungoffsets direkt erreicht werden. Den Unterprogrammen wird lediglich die Feldadresse, welche die Berechnungsdaten enthält, und die Anzahl der Daten zur Verfügung gestellt. Die eigentliche Berechnung erfolgt direkt mit dem Numerikprozessor 8087. Da alle Zwischenergebnisse im Coprozessor verbleiben, erreicht man eine hohe Berechnungsgeschwindigkeit.

Literatur

[1] Handbuch zu Turbo Pascal Vers. 3.0, Heimsoeth

[2] Die 8087/80287 numerischen Prozessor-Erweiterungen für 8086/80286 Systeme. te-wi Verlag München

Christian Neubauer

Einführung in Intuition

Teil 1: Screens und Windows

Da man als Computer-Anwender meist wenig Zeit hat, sich mit Fachliteratur eingehender zu befassen, und beim Amiga zusätzlich das Problem der Sprachbarriere auftritt, werden im Folgenden das Fenster-System und seine Werkzeuge – wie Gadgets oder Menüs – kurz gefaßt dargestellt.

Der Teil des Amiga-Betriebssystems, der für die Benutzeroberfläche zuständig ist, heißt Intuition, was andeuten soll, daß der Amiga intuitiv bedient werden kann. Alle Dinge, die mit Screens, Windows usw. zu tun haben, erledigt Intuition. Wie einfach man diese Funktionen auch als C-Programmierer nutzen kann, wissen allerdings die Wenigsten. Heute wird beschrieben, wie man auf dem Amiga Screens und Windows öffnet. Im zweiten Teil wird die Programmierung von Gadgets und Menüs das Hauptthema sein. Alle Programmbeispiele sind in C verfaßt, womit jeder die Möglichkeit hat, selbst Hand anzulegen und seine Programme in ein schönes Erscheinungsbild zu kleiden.

Funktionsaufrufe

Zum einen gibt es die Compilerfunktionen, die jeder in den Handbüchern des jeweiligen Compilers nachlesen kann. Hierzu zählen die Standardbibliothek und etliche „Amiga-Funktionen", wie etwa getcd() (get current directory) oder dfind() (find first directory entry). Andererseits existieren noch die „reinen" Amigafunktionen, die vom Betriebssystem zur Verfügung gestellt werden. Dieses wird beim Amiga 1000 bekanntlich von der Kickstart-Diskette geladen und steht bei den Modellen 500 und 2000 im eingebauten ROM zur Verfügung.

Libraries

Für diese Funktionen stellt der Amiga sogenannte Libraries – zu deutsch Bibliotheken – zur Verfügung. Eine solche Library ist eine Sammlung von Unterprogrammen, die mit einer großen Sprungtabelle beginnt. Den grundsätzlichen Aufbau dieser Tabelle zeigt *Bild 1.* Natürlich sind die Libraries nach Aufgabenbereichen geordnet. Folgende Libraries stehen zur Verfügung:

- clist.library
- console.library
- diskfont.library
- dos.library
- expansion.library
- exec.library
- graphics.library
- icon.library
- intuition.library
- layers.library
- mathffp.library
- mathieeedoubbas.library
- mathtrans.library
- potgo.library
- ram.library
- timer.library
- translator.library

Will man nun Funktionen einer Library aufrufen, so muß diese erst geöffnet werden. Beim Öffnen prüft das Betriebssystem, ob die betreffende Library schon im RAM vorhanden ist. Ist das nicht der Fall, so wird versucht, sie von Diskette bzw. Festplatte zu laden. Nachdem die Library im RAM steht, gibt Exec einen Vektor zurück, der auf die Basisadresse der Library zeigt. Hierbei nimmt Exec.library eine besondere Stellung ein: Sie ist nämlich immer über einen Zeiger an der absoluten Adresse 4 erreichbar – übrigens die einzige absolute Adresse im RAM des Amiga.

Hat man eine Library geöffnet, muß man noch den Offset der gewünschten Funktion nachschlagen (z. B. im Amiga Exec Manual), um dann endlich die Funktion aufrufen zu können. Wie in C eine Library geöffnet wird, zeigt *Bild 2.* Der OpenLibrary()-Funktion muß als Parameter der Name der Bibliothek und eine Versionsnummer angegeben werden. Diese Zahl steht für eine Libraryversion, die mindestens zur Verfügung stehen muß, damit der Aufruf nicht scheitert. Meistens gibt man hier 0 an, dann nämlich wird die verfügbare Version geöffnet.

Die OpenLibrary()-Funktion selbst steht in Exec.library und kann damit jederzeit aufgerufen werden.

```
00060A    JMP   $FC2FD6
000610    JMP   $FC0B28
000616    JMP   $FC0AC0
```

Bild 1. Aufbau der Sprungtabelle einer Library

```
#include <intuition.h>

struct IntuitionBase *IntuitionBase;

...

main()
{
IntuitionBase=(struct IntuitionBase *)
        OpenLibrary("intuition.library",0);
        if(IntuitionBase==NULL) exit(FALSE);
...
...
CloseLibrary(IntuitionBase);
}
```

Bild 2. Öffnen der Intuition.librarie

Intuition.library

Für uns ist in diesem Zusammenhang die Intuition.library interessant. In ihr stehen alle Funktionen, die benötigt werden, um Screens, Windows, Gadgets und Menüs zu programmieren. An dieser Stelle sei dem weitergehend interessierten Leser auch [1] empfohlen.

Bevor man jedoch Intuition-Funktionen benutzen kann, muß die Bibliothek geöffnet werden. Dabei erhält man den Vektor IntuitionBase, den man auch schleunigst retten sollte (Bild 2). Kann die Bibliothek, aus welchem Grund auch immer, nicht geöffnet werden, so gibt OpenLibrary() den Wert Null zurück.

Es sieht komplizierter aus, als es ist: Im Zeiger IntuitionBase wird der Rückgabewert der OpenLibrary()-Funktion gespeichert. Der Teil (struct IntuitionBase*) teilt dem Compiler lediglich mit, daß der Vektor, den OpenLibrary() liefert, nicht vom

Typ Integer, sondern vom Typ „Zeiger auf IntuitionBase" ist.
Geschafft! Ab sofort gewährt einem der Amiga Zugriff auf alle Funktionen von Intuition.

Aufgaben eines Screens

„Screens" sind die Grundlage von Intuition. Hat man z. B. die WorkBench geladen, so erscheinen die Icons (Diskettensymbole) auf einem Screen, der deshalb auch WorkBench-Screen genannt wird. Screens stellen die Umgebung für Bildschirm-Fenster dar und ermöglichen leichten Zugriff auf alle Bildschirm-Modi, also Interlace (= mit Zeilensprung), Hires (= hochauflösend), Sprites usw.
Screens erscheinen nach dem Öffnen immer in voller Größe, können jedoch verschoben werden – allerdings nur vertikal. Ein Screen hat immer drei „Gadgets", nämlich zwei in der rechten oberen Ecke, um den Screen in den Bildvorder- bzw. -hintergrund zu „senden" und die Titelleiste, die zur vertikalen Verschiebung dient.

Grundsätzlich muß zwischen zwei Arten von Screens unterschieden werden. Zum einen gibt es den WorkBenchScreen. Er ist immer da, weil mindestens ein Screen offen sein muß. Andererseits kann der Programmierer weitere, sogenannte „CustomScreens" öffnen. Hier lassen sich Auflösung, Darstellungsmodus für Farben usw. vor dem Öffnen einstellen.
CustomScreens wird man dann einsetzen, wenn man eben solche Darstellungsarten braucht, die der WorkBenchScreen nicht zur Verfügung stellt. In *Tabelle 1 und 2* sind die verschiedenen Darstellungsarten angegeben. Mit Tiefe ist die Anzahl der verwendeten Bits pro Bildpunkt gemeint.

Screen öffne Dich

Ist Intuition.library erst einmal geöffnet, kann man die OpenScreen()-Funktion benutzen. Diese erwartet als Parameter einen Zeiger auf eine „NewScreen-Struktur". Deren Aufbau und die Bedeutung ihrer Elemente zeigen *Bild 3* und *Tabelle 3*. Es muß also eine NewScreen-Struktur aufgebaut werden, was bei der Vielzahl der Einstellungen mitunter zu einem Problem werden kann. Der Screen wird dann mit folgender Programmzeile geöffnet:

```
Screen=(struct Screen *)OpenScreen
       (&NewScreen);
```

Dabei muß beachtet werden, daß NewScreen ein Zeiger auf eine solche Struktur

Tabelle 1: Die Auflösungen der verschiedenen Modes

Mode	Auflösung vertikal	Auflösung horizontal
LORES	200	320
HIRES	200	640
INTERL.	400	320

Tabelle 2: Farbenzahl und Speicherverbrauch für Lores-Screen

Tiefe	Farbenanzahl	Speicherverbrauch in kByte	verwendete Farb-Register
1	2	8	0...1
2	4	16	0...3
3	8	32	0...8
4	16	64	0...16
5	32	128	0...32

Tabelle 3: Die Bedeutung der NewScreen-Strukturelemente

LeftEdge, TopEdge	bestimmen die Koordinaten der oberen linken Ecke. Dabei gibt es allerdings einige Einschränkungen. LeftEdge wird momentan überhaupt nicht benutzt, sondern sollte, um Aufwärtskompatibilität zu späteren Versionen zu sichern, immer auf 0 gesetzt werden. Daraus folgt auch, daß Screens immer am linken Rand beginnen und nie in der Mitte des Bildschirms. TopEdge muß eine Zahl innerhalb der angegebenen Auflösung sein, d. h. zwischen 0 und 320 oder 0 und 640 liegen für LORES- bzw. HIRES-Sreens.
Width, Height	Diese beiden Elemente geben die Breite und die Höhe an. Auch hier muß eine Einschränkung beachtet werden. Die Summe von TopEdge und Height muß gößer sein als 256 bzw. 512, d. h. ein Screen muß immer bis zum unteren Rand des Bildschirms gehen, er kann nicht mittem im Bildschirm enden.
Depth	bestimmt die Anzahl der maximal darstellbaren Farben des Screens (Tabelle 2).
DetailPen, BlockPen	sind die Nummern der Farbregister, deren Inhalt für „Details", wie z. B. Gadgets, oder die Schrift in der Titelleiste, bzw. für „Blöcke", wie die Titelleiste selbst, benutzt werden.
ViewModes	ist eine SHORT Variable, in der einzelne Bits angeben, welche Displaymodes vom Programmierer gewünscht werden. Es stehen momentan fünf Modes zur Verfügung. Diese sind: HIRES Setzt man dieses Bit, so wird ein High-Resolution-Screen geöffnet, andernfalls ein LORES-Screen. INTERLACE Dieses Bit verdoppelt die vertikale Auflösung, indem der Amiga den Interlacemode benutzt. SPRITES ermöglicht die Benutzung von Sprites. DUALPF wird benutzt, wenn man den Amiga in den sog. DualPlayField Mode versetzen möchte. HAM ist die Abkürzung für HoldAndModifity. In diesem Modus kann der Amiga 4096 Farben gleichzeitig anzeigen.
Type	Für Type sind zwei Flags zulässig. Wir werden Type immer auf CUSTOMSCREEN setzen, was bedeudet, daß Intuition selbst Speicher für den Screen allokiert. Setzt man Type auf CUSTOMBITMAP, so benutzt Intuition den Speicher, auf den der Zeiger CustomBitMap zeigt (siehe weiter unten).
Font	Font ist ein Zeiger auf eine TextAttr-Struktur. In dieser Struktur ist festgelegt, welchen Zeichensatz Intuition für die Screenüberschrift benutzen soll. Setzt man dieses Feld auf NULL, so wird der Standardzeichensatz verwendet.
DefaultTitle	ist ein String (genaugenommen natürlich ein Zeiger auf einen String), der den „Namen" des Screens angibt.
Gadgets	Dieses Feld wird momentan nicht benutzt. Es sollte auf NULL gesetzt werden.
CustomBitMap	ist der oben erwähnte Zeiger auf eine Adresse, die Intuition als Startadresse für den Screenspeicher benutzen soll.

```
struct NewScreen
{
    SHORT LeftEdge, TopEdge, Width, Height, Depth;
    UBYTE DetailPen, BlockPen;
    USHORT ViewModes;
    USHORT Type;
    struct TextAttr *Font;
    UBYTE *DefaultTitle;
    struct Gadget *Gadgets;
    struct BitMap *CustomBitMap;
};
```

Bild 3. Aufbau der NewScreen-Struktur

```
struct NewWindow
{
    SHORT LeftEdge, TopEdge;
    SHORT Width, Height;
    UBYTE DetailPen, BlockPen;
    ULONG IDCMPFlags;
    ULONG Flags;
    struct Gadget *FirstGadget;
    struct Image *CheckMark;
    UBYTE *Title;
    struct Screen *Screen;
    struct BitMap *BitMap;
    SHORT MinWidth, MinHeight;
    USHORT MaxWidth, MaxHeight;
    USHORT Type;
};
```

Bild 4. Aufbau der NewWindow-Struktur

und nicht etwa die Struktur selbst ist. OpenScreen() gibt einen Zeiger zurück, der auf eine Screen-Struktur zeigt. Auch dieser Zeiger muß gerettet werden, da man den Screen sonst nicht mehr schließen und damit den von ihm belegten Speicher nie mehr freigeben kann.

Die Screen-Struktur

Die Screen-Struktur verdient noch etwas Aufmerksamkeit, denn sie enthält einige interessante Daten: Vorne an steht hier 'TopEdge'. In dieser Short-Variable steht, auf welche Y-Koordinate der Benutzer den Screen gezogen hat. In den Variablen 'MouseX' und 'MouseY' – ebenfalls vom Typ Short – stehen die Koordinaten des Mauszeigers relativ zur linken, oberen Ecke des Screens. Für Profis, die direkt auf den Screen schreiben wollen, sind die Elemente 'ViewPort', 'RastPort', 'BitMap' und 'LayerInfo' wichtig, deren genaue Bedeutung hier jedoch nicht besprochen werden kann. Drei Funktionen, die sich auf Screens beziehen, sind hier unbedingt zu nennen: Mit ihnen kann man das ausführen, was der Benutzer mit den Screen-Gadgets und der Maus auch machen kann. Sie heißen:

```
MoveScreen(Screen,DeltaX,DeltaY);
ScreentoBack(Screen);
ScreentoFront(Screen);
```

Dabei ist „Screen" der Zeiger auf die Screen-Struktur. Am Ende muß der Screen natürlich wieder geschlossen werden. Dies macht man mit der Funktion CloseScreen(Screen); – diese Funktion baut den Screen wieder ab, gibt den Speicher der Bitmap frei und befreit das System von den für die Verwaltung notwendigen Aufgaben. Mit einem Screen alleine ist es jedoch noch nicht getan. Um die Kommunikation mit dem Benutzer herzustellen, ist schon ein Window (Bildschirm-Fenster) nötig.

```
#include <intuition/intuition.h>
#include <string.h>

struct IntuitionBase *IntuitionBase;
struct Screen *Screen;
struct Window *wdw1, *wdw2;

struct IntuiText Text =
{
 1,0,                    /*FrontPen,BackPen              */
 JAM2,                   /*DrawMode                      */
 0,0,                    /*LeftEdge,TopEgde              */
 NULL,                   /*Standard-Zeichensatz          */
 "Text in 16 Farben !",  /*Text                          */
 NULL                    /*keine weiteren IText-strukts */
};

struct NewScreen NewScreen =
{
 0,0,640,254,4,          /*LeftEdge,TopEdge,Width,Height,Depth*/
 0,1,                    /*DetailPen,BlockPen                  */
 HIRES,                  /*Hires-Screen                        */
 CUSTOMSCREEN,           /*Type                                */
 NULL,                   /*Standard-Zeichensatz benutzen       */
 "My HiRes-Screen",      /*Titel                               */
 NULL,                   /*Gadgetfeld muss NULL gesetzt werden*/
 NULL                    /*Intuition soll BitMap organisieren */
};

struct NewWindow NewWdw1 =
{
 0,12,210,180,           /*LeftEdge,TopEdge,Width,Height       */
 0,1,                    /*DetailPen,BlockPen                  */
 NULL,                   /*wdw1 erhaelt keine Messages         */
 WINDOWDRAG|WINDOWDEPTH|       /*Bedeutung der Flags s.Text*/
 SMART_REFRESH,
 NULL,NULL,              /*FirstGadget,CheckMark auf NULL      */
 "My Color Window",      /*Titel des Windows                   */
 NULL,NULL,              /*Screen,BitMap auf NULL              */
 0,0,0,0,                /*Mini-und Maximale Groesse           */
 CUSTOMSCREEN            /*Window nicht auf WBenchScreen       */
};

struct NewWindow NewWdw2 =
{
 300,75,180,10,          /*LeftEdge,TopEdge,Width,Height       */
 0,1,                    /*DetailPen,BlockPen                  */
 CLOSEWINDOW|MOUSEMOVE,        /*IDCMPFLAGS                    */
 WINDOWDRAG|WINDOWCLOSE|       /*Bedeutung der Flags s.Text*/
 REPORTMOUSE|SMART_REFRESH|
 ACTIVATE,
 NULL,NULL,              /*FirstGadget,CheckMark auf NULL      */
 NULL,                   /*Titel des Windows                   */
 NULL,NULL,              /*Screen,BitMap auf NULL              */
 0,0,0,0,                /*nicht veraenderbar                  */
 CUSTOMSCREEN            /*Window nicht auf WBenchScreen       */
};

void main()
long i;
struct IntuiMessage *message;
```

Bild 5. Dieses Programm öffnet auf dem Amiga einen CustomScreen mit zwei Fenstern: Im ersten Fenster wird mit 16 Farben geschrieben; das zweite Fenster besteht nur aus einer Kopfleiste mit der Cursorposition relativ zur oberen rechten Ecke dieses Fensters

```
    ULONG class;
    USHORT code, x, y;
    char *outx="       ",*outy="       ";

    /* Erst Intuition.library oeffnen */

    IntuitionBase = (struct IntuitionBase *)
       OpenLibrary("intuition.library",0);
            if(IntuitionBase==NULL) exit(200);

    /* Dann Screen oeffnen              */

    Screen=(struct Screen *) OpenScreen(&NewScreen);
    if(Screen==NULL) exit(100);

    /* Screenpointer in die NewWindow-Struktur */
    /* schreiben                               */

    NewWdw1.Screen=Screen;
    NewWdw2.Screen=Screen;

    /* Jetzt die zwei Windows aufmachen */

    wdw1=(struct Window *) OpenWindow(&NewWdw1);
    if(wdw1==NULL) exit (50);

    wdw2=(struct Window *) OpenWindow(&NewWdw2);
    if(wdw2==NULL) exit (50);

    /* Text in wdw1 schreiben, dabei FrontPen  */
    /* veraendern                              */

    for(i=1;i<17;i++)
    {
     Text.FrontPen=i;
     Text.TopEdge=i*10;
     PrintIText(wdw1->RPort,&Text,10,0);
    }

    /* Auf Messages warten. Sobald eine ankommt      */
    /* abfragen welche Msg. ankam und reagieren.     */
    /* Bei CLOSEWINDOW alles schliessen und beenden */
    /* Bei MOUSEMOVE Koordinatem holen und ausgeben */

    for(;;)
    {
       Wait(1<<wdw2->UserPort->mp_SigBit);

       while(message=(struct IntuiMessage *)
                     GetMsg(wdw2->UserPort))

         {
          class=message->Class;  /* Parameter retten  */
          code= message->Code;
          x=    message->MouseX;
          y=    message->MouseY;

          ReplyMsg(message);      /* Msg. zurueckgeben */
         }
          if(class==MOUSEMOVE)       /* Bei MOUSEMOVE      */
              {
               Text.LeftEdge=0;
               Text.TopEdge=0;
               Text.FrontPen=10;
               Text.BackPen=
               stci_d(outx,x
               stci_d(outy,y);
               Text.IText="                ";
               PrintIText(wdw2->RPort,&Text,29,1);
               Text.IText=outx;
               PrintIText(wdw2->RPort,&Text,29,1);
               Text.IText=outy;
               PrintIText(wdw2->RPort,&Text,100,1);
              }

          if(class==CLOSEWINDOW)   /* Bei CLOSEWINDOW    */
              {
               CloseWindow(wdw1);
               CloseWindow(wdw2);
               CloseScreen(Screen);
               CloseLibrary(IntuitionBase);
               exit(0);
              }
       }
}
```

Windows

Ähnlich wie bei Screens muß auch bei den Fenstern zunächst eine Struktur mit den Parametern gebildet werden. Den Aufbau dieser Struktur zeigt *Bild 4.* Die Bedeutung der Elemente ist in *Tabelle 4* zusammengefaßt. Die ersten sechs Elemente haben die gleiche Bedeutung wie die der NewScreen-Struktur. Doch einige andere Elemente verdienen eine eingehendere Besprechung.

Der IDCMPPort

Spätestens bei IDCMPFlags kommt man um eine nähere Betrachtung des Betriebssystem-Aufbaus nicht herum. Das Konzept des Amiga beruht auf 'Messages', zu deutsch also Meldungen. Verschiedene Teile des Betriebssystems verständigen sich untereinander über solche Meldungen und teilen damit das Eintreten bestimmter Ereignisse mit. Auch Intuition tut das. Es können jedoch nicht nur andere Betriebssystemteile Messages erhalten, sondern auch Anwenderprogramme. Wie man sich leicht vorstellen kann, gibt es zahlreiche Messages, z. B. die Meldung, daß der Anwender das CloseGadget (bei vielen Windows links oben) betätigt hat, oder daß das Bildschirm-Fenster in seiner Größe verändert wurde.
Jetzt ist es an der Zeit zu verraten, was IDCMP überhaupt heißt: Intuition Direct Communications Message Port. Lesern, die sich unter so einen MessagePort noch nichts rechtes vorzustellen können, sei gesagt, daß das System ein Langwort zur Verfügung stellt, in dem jedes Bit einem bestimmtem Ereignis zugeordnet wurde. Ist ein Bit gesetzt, so ist dieses Ereignis eingetroffen.
Um auf unsere NewWindow-Struktur zurückzukommen: In der ULONG-Variable IDCMPFlags kann der Programmierer festlegen, welche Meldungen er erhalten möchte und welche Intuition für sich behalten soll. Den Ereignissen, die ja nur durch bestimmte Bits vertreten sind, wurden per '#define'-Anweisung Namen zugewiesen. Dies wird in der Header-Datei intuition.h getan. Wer also einmal alle Messages sehen will, soll dort nachsehen. Für unsere Zwekke sind nur folgende Messages wichtig:

Mousemove
teilt mit, daß die Maus bewegt wurde.

Gadgetup
ist für die Arbeit mit Gadgets wichtig.

Menupick
wird bei Menüs gebraucht.

Tabelle 4: Beschreibung der NewWindow-Struktur-Elemente

LeftEdge, TopEdge, Width, Height	sind X- und Y-Koordinate der oberen linken Ecke des Windows. Width und Height geben die Breite und die Höhe an.
DetailPen, BlockPen	haben die gleiche Bedeutung wie bei den Screens (s. dort).
IDCMPFlags	ausführliche Beschreibung siehe Text.
Flags	ausführliche Beschreibung siehe Text.
FirstGadget	ist ein Zeiger, der auf eine verkettete Liste von Gadget-Strukturen zeigt, wenn man Gadgets im Fenster benutzt. Diesen Zeiger werden wir nächstes Mal brauchen. Setzt man FirstGadget auf NULL, so nimmt Intuition an, daß das Fenster keine Gadgets hat.
CheckMark	ist ein Zeiger auf eine Image-Struktur. Mit ihm kann man das Aussehen des „Häkchens", das bei Menüs ab und zu auftaucht, verändern. CheckMark wird fast immer auf NULL gesetzt.
Title	zeigt auf einen mit 0 beendeten String, den Intuition in die Titelzeile schreibt.
Screen	Beschreibung siehe Text.
BitMap	Benutzt man das Fenster für grafische Zwecke, so kann man hier die Adresse seiner eigenen BitMap für das Fenster angeben. Üblicherweise ist auch dieses Feld NULL.
MinWidth, MinHeight, MaxWidth, MaxHeight	ausführliche Beschreibung siehe Text.
Type	Muß entweder auf WBENCHSCREEN oder auf CUSTOMSCREEN gesetzt werden (s. Text).

Closewindow
besagt, daß das CloseWindow-Gadget betätigt wurde.

Wer noch nicht ganz durchschaut, möge sich das Programm ansehen, in dem das alles in die Praxis umgesetzt wurde.

Die Flags

Als nächstes Element fällt 'Flags' auf. Mit Hilfe dieser Flags teilt man Intuition mit, welche Eigenschaften das Fenster haben soll. Hier gibt es einige Möglichkeiten:

- WINDOWDRAG erlaubt es, das Fenster zu bewegen
- WINDOWDEPTH versieht das Fenster mit den „TiefenGadgets" (oben rechts)
- WINDOWCLOSE sorgt für ein Close-Gadget
- WINDOWSIZING erlaubt es, die Größe zu verändern

Einer weitergehenden Erläuterung bedürfen die Flags SIMPLE_REFRESH und SMART_REFRESH. Diese beiden teilen Intuition mit, wie der 'Refresh' des Fensters durchgeführt werden soll, d. h. was gemacht werden soll, wenn Teile des Fensters verdeckt waren und anschließend wieder freigegeben wurden. Gibt man SIMPLE_REFRESH an, so macht Intuition schlicht gar nichts: Was einmal verdeckt war, wird nicht wieder hergestellt. Bei SMART_REFRESH muß man sich nur um den Bildaufbau kümmern, wenn das Fenster größer gemacht wird. Das Flag REPORTMOUSE sorgt dafür, daß Mausbewegungen an das dem Fenster zugeordnete Programm übermittelt werden. Interessant ist auch das Flag BACKDROP, das dem Window die Eigenschaft verleiht, immer das unterste aller auf dem Screen befindlichen zu sein. Braucht man ein Fenster ohne Rand, so gibt man BORDERLESS an – das Window hat dann weder Rand noch Titelleiste. Dies gilt allerdings nur unter der Bedingung, daß das Fenster keine System-Gadgets hat. Um sein Window gleich zum aktiven Fenster zu machen, setzt man ACTIVATE. Als letztes Flag muß noch NOCAREREFRESH genannt werden, das man immer setzen sollte, wenn man von RefreshMessages seitens Intuition verschont bleiben will.

Weitere Elemente

Im Struktur-Element Screen muß der Zeiger, der auf die Screen-Struktur zeigt, eingesetzt werden. Dieser ist im allgemeinen aber beim Aufbau der NewWindow-Struktur noch nicht bekannt und wird deshalb vom Hauptprogramm nachträglich eingesetzt. Die Variablen MinWidth, MinHeight, MaxWidth und MaxHeight sind die kleinsten und die größten Maße, auf die das Fenster durch das SizingGadget gebracht werden kann. Gibt man hier jeweils 0 an, so ist die Größe nicht veränderbar. Um das Window auf dem WorkBench-Screen zu öffnen, muß man bei Type WBENCHSCREEN angeben, andernfalls CUSTOMSCREEN.

Der letzte Schritt

Ist die NewWindow-Struktur initialisiert, dann öffnet man das Fenster mit:

```
Window=(struct Window *)
        OpenWindow(&NewWindow);
```

Wenn alles richtig war, erscheint dann das Fenster auf dem entsprechenden Screen. Wie immer muß man den erhaltenen Zeiger retten, der diesmal auf eine Window-Struktur zeigt, um später das Fenster wieder schließen zu können. Von Nutzen sind in der Window-Struktur nur die Elemente LeftEdge, TopEdge, Width und Height, die alle die tatsächlichen Werte darstellen. (Der Benutzer kann ja die ursprüngliche Position und Größe verändern.) Auch mit dem Fenster lassen sich programmgesteuert einige Dinge machen. Die Funktionen sprechen weitestgehend für sich und werden deshalb nicht weiter erläutert:

```
– MoveWindow(Window,DeltaX,DeltaY);
– SizeWindow(Window,DeltaX,Deltay);
– WindowtoFront(Window);
– WindowtoBack(Window);
```

Das Programm

Es ist in jedem Fall empfehlenswert, das Listing in *Bild 5* wenigstens einmal anzusehen. Mit ihm werden wahrscheinlich die restlichen Unklarheiten beseitigt. Für die, die es nicht abtippen wollen, sei gesagt, daß das Programm einen Screen mit Tiefe 4 und darauf zwei Windows öffnet, bis der Benutzer das CloseGadget betätigt. Das eine zeigt, wenn es das aktive ist, die aktuellen Mauszeiger-Koordinaten relativ zur oberen linken Ecke des Fensters an. In das andere wird Text mit 16 Farben geschrieben. Das soll's für heute gewesen sein. Basteln Sie ruhig etwas mit Windows herum, auch wenn dabei der Guru des Öfteren meditiert. Das ist am Anfang ganz normal und es sollte sich dadurch keiner entmutigen lassen. Viel Spaß bis zum nächsten Mal!

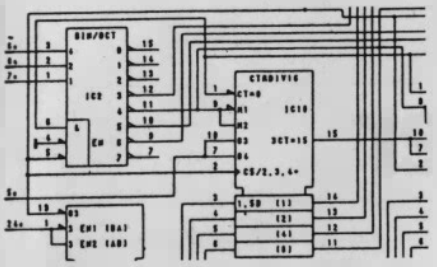
(Wird fortgesetzt)

Literatur

[1] Amiga Intuition Reference Manual, Addison Wesley, Reading, Massachusetts, 1986
[2] Amiga Rom Kernel Reference Manual, Addison Wesley, Reading, Massachusetts, 1986
[3] Amiga Intern, Data Becker, Düsseldorf, 1987

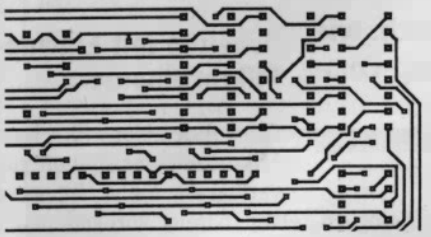
Mit SHAMROCK SOFTWARE können Sie zum Beispiel ...

...Schaltbilder zeichnen,



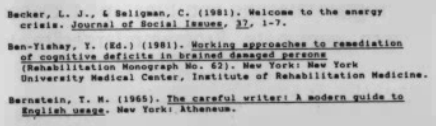
SHAMROCK-CAD ist dafür prädestiniert: Eine große Bibliothek an Standard-Schaltzeichen (z. B. TTL-Bausteine, diskrete Bauelemente nach DIN) wird mitgeliefert. Bibliotheken können nach Bedarf erstellt, erweitert oder modifiziert werden. Ganze Zeichnungsbereiche können verschoben, gelöscht oder kopiert werden. Die Zeichnung kann auf Plotter oder Drucker ausgegeben werden. Die Bedienung erfolgt über Tastatur und/oder (Microsoft-kompatible) Maus. Ein Signal-Analyse-Programm stellt die Verbindung zum AUTOROUTER 3 her ..... **495 DM**

Platinen-Layouts erstellen,



AUTOROUTER 3 ermöglicht die interaktive Plazierung der Bauelemente auf einer ein- oder zweiseitigen Platine bis zur Größe einer Doppel-Europakarte. Verbindungen können manuell vorverlegt oder automatisch mit Hilfe eines hervorragend optimierten Routing-Algorithmus erstellt werden, der beispielsweise Busstrukturen bevorzugt behandelt. Eine von SHAMROCK-CAD aus dem Schaltplan erstellte Signal-Verbindungsliste kann direkt verarbeitet werden ..... **764 DM**

Literatur verwalten,



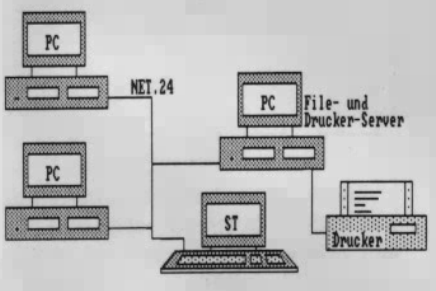
Becker, L. J., & Seligman, C. (1981). Welcome to the energy crisis. Journal of Social Issues, 37, 1-7.

Ben-Yishay, Y. (Ed.) (1981). Working approaches to remediation of cognitive deficits in brained damaged persons (Rehabilitation Monograph No. 62). New York: New York University Medical Center, Institute of Rehabilitation Medicine.

Bernstein, T. M. (1965). The careful writer: A modern guide to English usage. New York: Atheneum.

Das Programm BRAIN wurde – ohne die Kompromisse einer Universal-Datenbank – speziell für die Verwaltung von Literaturstellen konzipiert: Menüführung, variable Feldlänge, eingebauter Editor für Abstracts, komfortable Suche mit vielen Verknüpfungsmöglichkeiten, vielfältige Ausgabeformate, Auslagern und Mischen von Dateien ..... **425 DM**

PCs vernetzen ...



NET.24 ist eine preiswerte Lösung für Büro-Kommunikation und Daten-Transfer für bis zu 26 PCs. Je PC ist dafür nur eine serielle Schnittstelle nötig. RAM-residente Software, im Hintergrund laufender Server (zentrale Dateien, gemeinsamer Drucker). NET.24 ist etwas völlig anderes als das, was man sich gemeinhin unter einem LAN (Local Area Network) vorstellt. Es realisiert nicht nur den obligatorischen Datei-Transfer, sondern schafft auch ein neues Kommunikationsmittel. Starter-Kit für zwei PCs inkl. Server und 5 m Kabel **303.98 DM**

... und vieles andere mehr:

PCMAIL, die universelle Mailbox ..... **485 DM**
PCTERM, das komfortable Terminal-Programm mit Up-/Download ..... **98 DM**
TELESERVICE, die Fernwartungs-Software für Systemhäuser ..... **485 DM**
EDI, der schnelle Full-Screen-Editor für alle EDLIN-Geplagten ..... **98 DM**
V.24-OEM, Gerätetreiber für COM1...COM4, 110...38 400 Bd ..... **198 DM**
MS-DOS-UTILITY zum Konvertieren von Daten, Drucker umleiten usw. ..... **98 DM**

shamrock software

Und das ist noch keineswegs alles! Bitte fordern Sie unseren kostenlosen Gesamtkatalog an.
(Alle genannten Programme sind auf Industriestandard-kompatiblen MS-DOS-Rechnern lauffähig.)

SHAMROCK SOFTWARE Vertrieb GmbH

Karlstraße 35, 8000 München 2, Telefon 0 89/59 54 68

Franz-Josef Katgely

mc-EPROM-Brenner für Einchip-Computer

Die mc-EMUFs sind so erfolgreich, weil man gängige Tischcomputer als Entwicklungssystem verwenden kann. Jetzt bekommen sie Konkurrenz aus dem eigenen Haus: Wir zeigen Ihnen nämlich, wie man mit dem mc-EPROM-Brenner [1] Einchip-Computer der MCS48-Familie programmiert.

Zwei typische Vertreter der MCS48-Familie, der 8748 und der 8749, die sich für die meisten Anwendungen eignen, haben 24 Ein-/Ausgabe-Leitungen, 1 bzw. 2 KByte Progammspeicher in Form eines EPROMS, 64 bzw. 128 Byte Datenspeicher (RAM), eine 8-Bit-CPU, einen Timer, einen Taktgenerator auf dem Chip und einen speziell auf Ein-/Ausgabe-Aufgaben abgestimmten Befehlssatz. Alle Assemblerbefehle werden in ein oder zwei Maschinenzyklen ausgeführt und garantieren dadurch eine schnelle Abarbeitung der einzelnen Befehle. Die Kosten für einen solchen Chip liegen für die wiederlöschbare Version mit Quarzfenster unter 30 DM und für die nur einmal programmierbare OTP-Version (One Time Programmable) im Plastikgehäuse ohne Quarzfenster unter 10 DM. Gerade der erstaunlich günstige Preis macht diese Chips für viele Anwender überaus interessant.

Arbeiten mit Einchip-Mikrocomputern

Cross-Assembler für Einchip-Mikrocomputer sind oft günstig verfügbar. Meist ist es auch möglich, einen bereits vorhandenen (Makro-)Assembler (z. B. MASM) so „umzustricken", daß er MCS48-Code erzeugt. Im Gegensatz zu Cross-Assemblern sind zur Programmierung von Einchip-Mikrocomputern geeignete Programmiergeräte leider

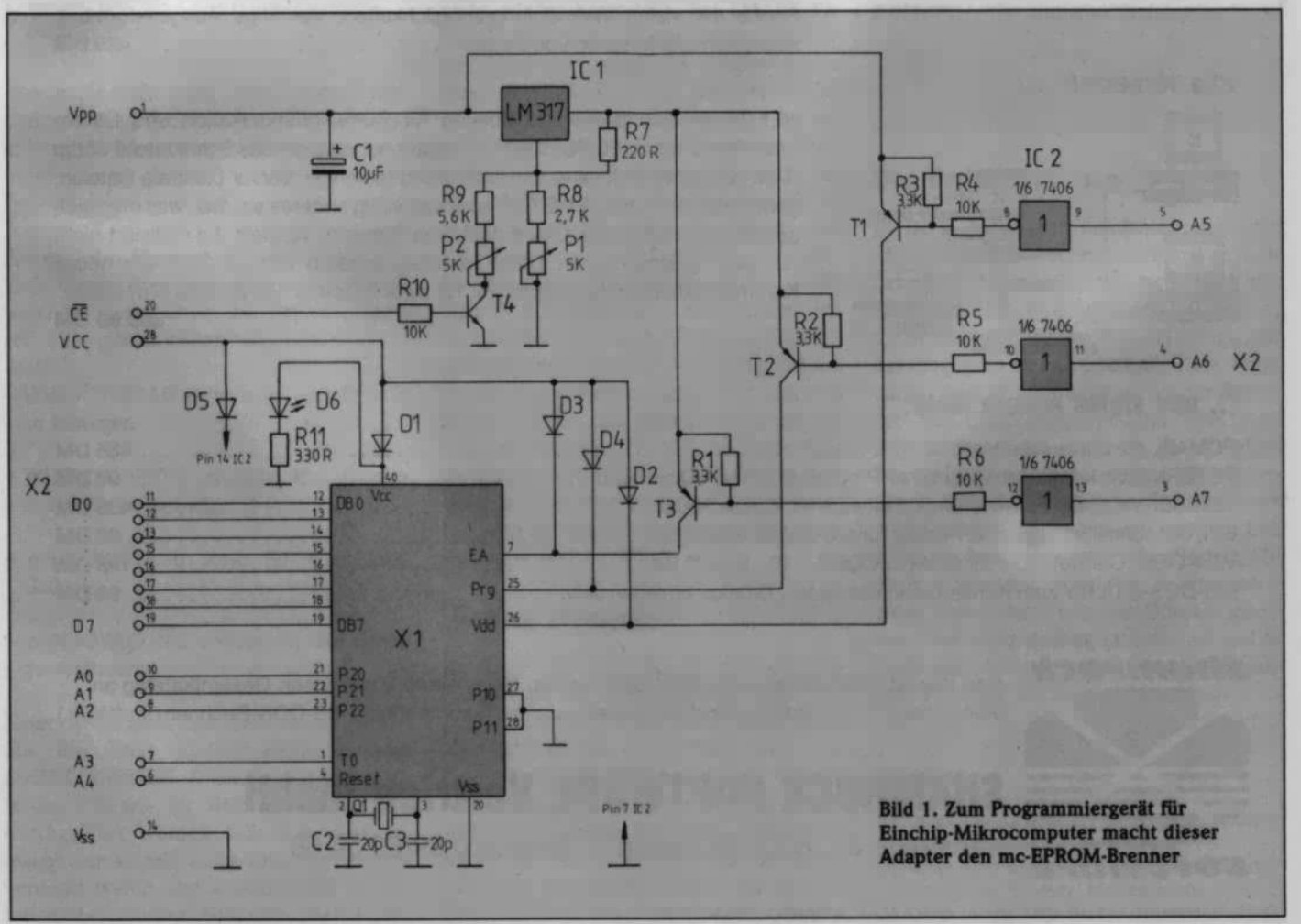


Bild 1. Zum Programmiergerät für Einchip-Mikrocomputer macht dieser Adapter den mc-EPROM-Brenner

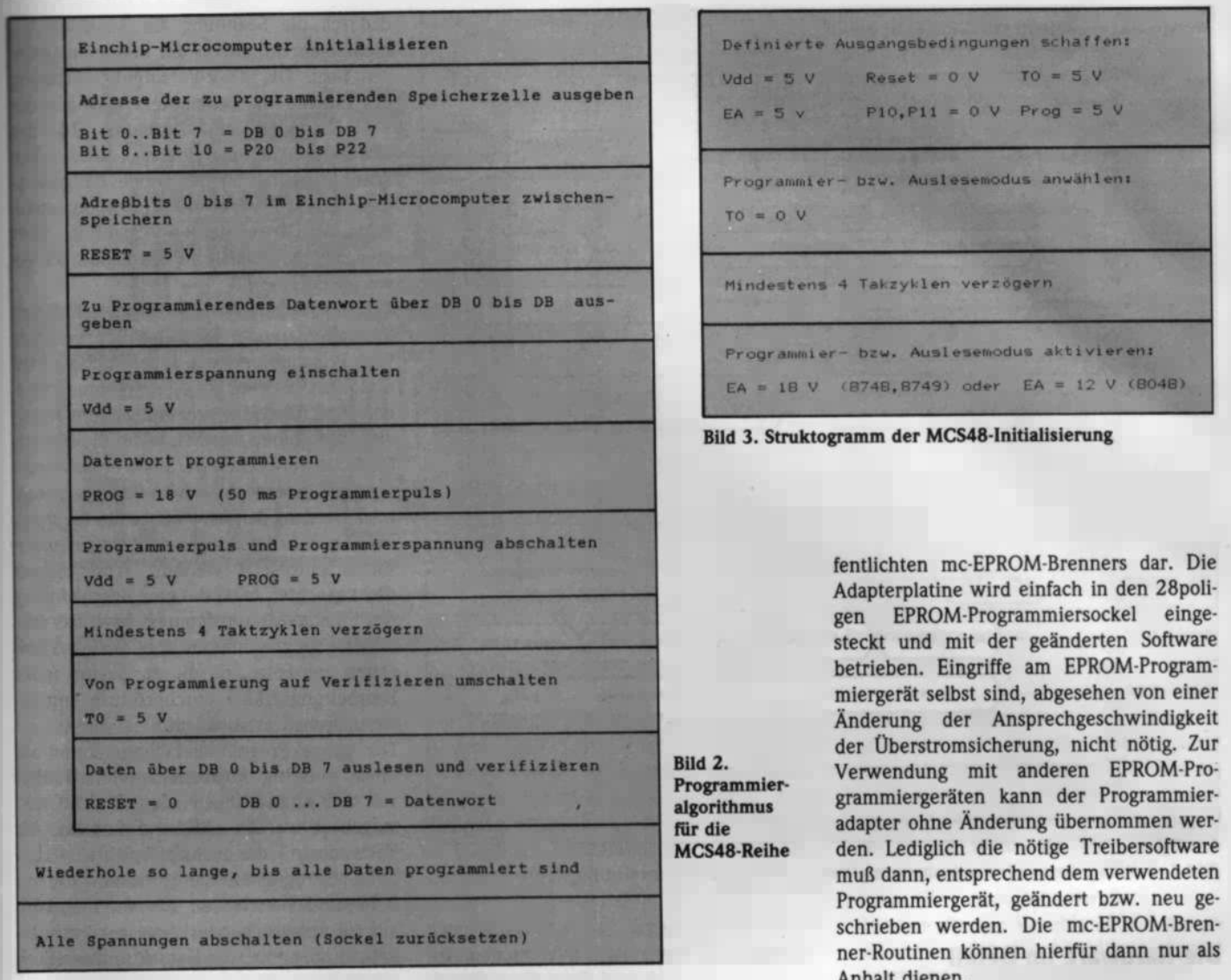


Bild 2. Programmieralgorithmus für die MCS48-Reihe

Bild 3. Struktogramm der MCS48-Initialisierung

nur zu haarsträubenden Preisen erhältlich, und oft sind diese Geräte dann nur ausschließlich zur Programmierung von Einchip-Mikrocomputern geeignet, d. h. es ist nicht möglich, gängige EPROMS damit zu programmieren. Die Preise für solche Geräte liegen im allgemeinen jenseits der 500-Mark-Schmerzschwelle. Billiger ist die Zusatzschaltung im *Bild 1*.

Ein Wort zu MCS48-Cross-Assembler

Im allgemeinen erzeugen diese Cross-Assembler als Ausgabedatei eine Datei im Intel-Hex-Format. Weil es sich hierbei um eine ASCII-Datei mit Adreßinformationen und Prüfsummen handelt, kann sie nicht direkt in den Programmspeicher der Einchip-Mikrocomputer programmiert werden. Diese Hex-Datei muß in eine Objekt-Datei umgewandelt werden. Ein für solche Umwandlungen gut geeignetes Turbo-Pascal-Programm wurde in [2] vorgestellt. Dieses Programm wandelt die Hex-Datei in eine direkt in den Einchip-Mikrocomputer programmierbare Objekt-Datei um.

Zur Programmierung von MCS48-Chips

Diese Einchip-Mikrocomputer müssen mit einem ganz speziellen Programmieralgorithmus programmiert werden und benötigen während der Programmierung zum Teil ungewöhnliche Spannungspegel. Der genaue Ablauf des Programmiervorgangs kann aus den in *Bild 2* und *Bild 3* abgedruckten Struktogrammen entnommen werden. Ferner wird auf das in *Bild 4* gezeigte Programmier-Timing-Diagramm hingewiesen.

Der Programmieradapter

Der beschriebene Progammieradapter (Bild 1) stellt eine Erweiterung des in [1] veröffentlichten mc-EPROM-Brenners dar. Die Adapterplatine wird einfach in den 28poligen EPROM-Programmiersockel eingesteckt und mit der geänderten Software betrieben. Eingriffe am EPROM-Programmiergerät selbst sind, abgesehen von einer Änderung der Ansprechgeschwindigkeit der Überstromsicherung, nicht nötig. Zur Verwendung mit anderen EPROM-Programmiergeräten kann der Programmieradapter ohne Änderung übernommen werden. Lediglich die nötige Treibersoftware muß dann, entsprechend dem verwendeten Programmiergerät, geändert bzw. neu geschrieben werden. Die mc-EPROM-Brenner-Routinen können hierfür dann nur als Anhalt dienen.

Änderung an der mc-EPROM-Brenner-Hardware

Leider läßt es sich nicht umgehen, die Ansprechgeschwindigkeit der Überstromsicherung des mc-EPROM-Brenners herabzusetzen. Dies geschieht dadurch, daß C10 des mc-EPROM-Brenners durch einen Kondensator von 5 µF ersetzt wird. Hierdurch wird zwar die Empfindlichkeit der Überstromsicherung vermindert, jedoch ist immer noch ein in den meisten Fällen ausreichender Schutz des zu programmierenden Chips gewährleistet. Hier sollte der Anwender sich jedoch nicht scheuen, eigene Versuche anzustellen, um den für seine Anordnung optimalen Wert von C10 zu ermitteln. Der Wert von 5 µF stellt den in der Regel optimalen Kompromiß zwischen hoher Ansprechgeschwindigkeit (Empfindlichkeit der Überstromsicherung) und hoher Störsicherheit (Sicherheit gegen zu sensibles Verhalten der Überstromsicherung bei der Programmierung) dar.

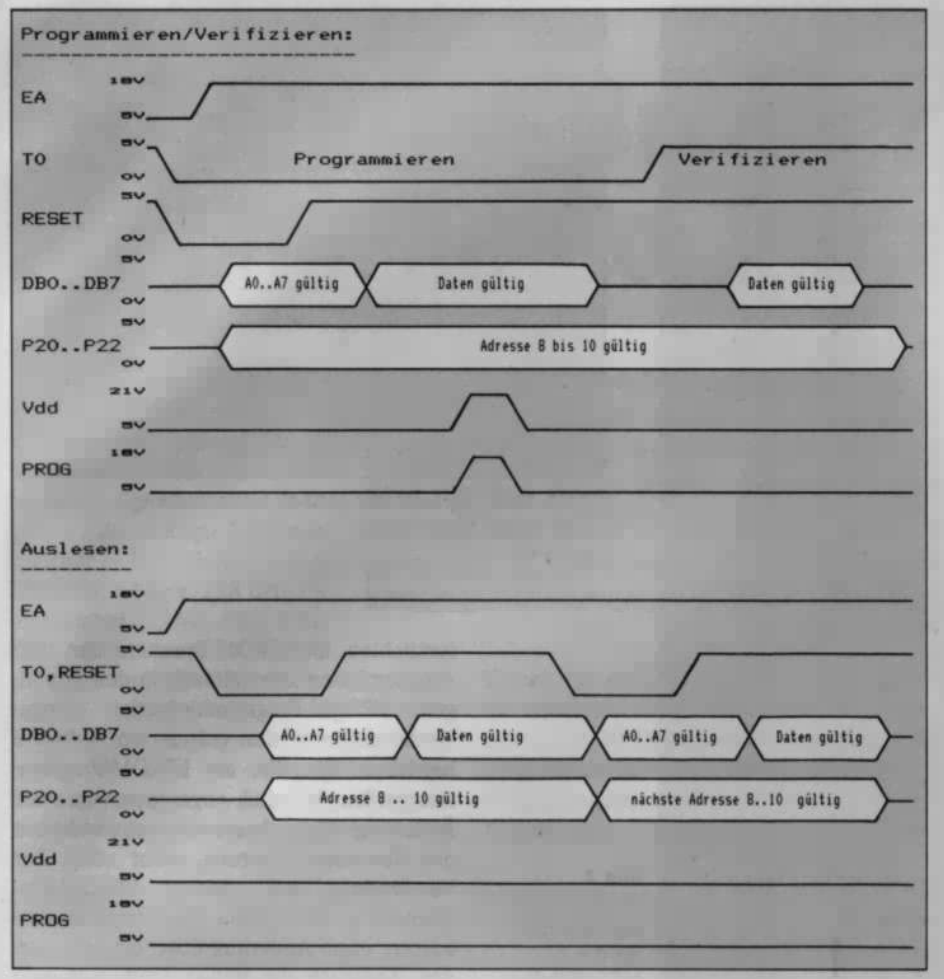


Bild 4. Zeitdiagramme für Programmieren/Verifizieren und Auslesen

Die Hardware im Detail

Alle Anschlußbezeichnungen des EPROM-Programmiergerätes beziehen sich auf die Anschlußbelegung des Typs 2764.

Die zur Bearbeitung des Typs MCS48 benötigten Spannungspegel werden durch IC1 (LM 317) erzeugt. Als Eingangsspannung für dieses IC dient die über Anschluß 1 des EPROM-Programmiersockels eingespeiste Programmspannung von 21 V. Mit Hilfe des Transistors T4 wird die Ausgangsspannung von IC1 zwischen 12 V (Auslesen/Verifizieren des 8048) und 18 V (Programmieren/Verifizieren/Auslesen des 8748 bzw. 8749) umgeschaltet. T4 erhält sein Eingangssignal über den Anschluß 20 des EPROM-Programmiergerätes (CE-Anschluß am EPROM-Programmiergerät).

Die Signale für die Anschlüsse V_{dd}, EA und Prog des Einchip-Mikrocomputers werden über die Transistoren T1...T3 an die entsprechenden Anschlüsse des Einchip-Programmiersockels X1 geschaltet. Die Transistoren erhalten ihr Eingangssignal aus OC-Invertern von IC2 (7406). Die Dioden D2...D4 verhindern, daß bei Signalspannungen von mehr als 5,5 V an den Anschlüssen V_{dd}, EA und Prog diese Signalspannungen über die +5-V-Versorgungsspannung kurzgeschlossen werden.

Die Datenleitungen D0...D7 des EPROM-Programmiergerätes werden direkt mit den entsprechenden Anschlüssen des Einchip-Mikrocomputer-Programmiersockels verbunden. Ebenso werden die Adreßleitungen A0...A2 des EPROM-Programmiergerätes direkt an die Anschlüsse P20...P22 des Programmiersockels angeschlossen.

Die Adreßleitungen A3...A4 des Programmiergerätes werden an die Anschlüsse T0 und RESET des Einchip-Mikrocomputer-Programmiersockels geführt.

Die Anschlüsse P10 und P11 müssen laut Datenblatt der Firma Intel auf Massepegel liegen.

Der zu programmierende Einchip-Mikrocomputer wird über Anschluß 28 des EPROM-Programmiergerätes mit Betriebsspannung versorgt. Die Spannung am Anschluß 28 des EPROM-Programmiergerätes muß 6 V betragen, da die Rückstrom-Schutzdiode D2, je nach Stromfluß, einen erhöhten Spannungsabfall verursacht und dadurch die Spannung am Anschluß V_{dd} nicht mehr innerhalb der erlaubten Grenzen liegt. Die +5-V-Versorgungsspannung des Einchip-Mikrocomputers wird aus der +6-V-Spannung am Anschluß 28 des EPROM-Programmiergerätes über den Spannungsabfall an der Diode D1 gewonnen. Ebenso wird die Versorgungsspannung für IC2 aus der +6-V-Spannung über den Spannungsabfall an der Diode D5 gewonnen.

Die Leuchtdiode D6 signalisiert, daß der zu programmierende Einchip-Mikrocomputer vom Programmiergerät angesprochen wird und nicht aus dem Sockel entnommen werden darf. Der Mikrocomputer kann jederzeit entnommen werden, wenn die Leuchtdiode D6 verloschen ist. Sollte er bei leuchtender Diode doch versehentlich entnommen werden, darf er, solange D6 leuchtet, auf keinen Fall wieder in den Programmiersockel eingesetzt werden! Wird hierauf nicht geachtet, hätte das eine Beschädigung des Einchip-Mikrocomputers bzw. das fehlerhafte Programmieren oder Auslesen von Daten zur Folge, da die zu Beginn jedes Bearbeitungszyklus durchgeführte Initialisierung unwirksam würde.

Der ganze Programmieradapter kann auf einer einseitig kupferkaschierten Platine mit den Abmessungen 75 mm × 85 mm aufgebaut werden (*Bilder 5* und *6*). Als Verbindung der Adapterplatine zum EPROM-Programmiergerät-Sockel können 2,54-mm-Steckerleisten dienen, die dann auf die entsprechende Länge gekürzt werden. Hier sollten, um Kontaktschwierigkeiten zu vermeiden, nach Möglichkeit Steckleisten mit vergoldeten Kontakten Anwendung finden.

Als Programmiersockel (X1) sollte unbedingt ein Nullkraftsockel, am besten ein 40poliger Textool-Sockel, verwendet werden.

Abgleich des Programmieradapters

Setzen Sie während des Abgleiches keinen Chip in die Fassung des Programmieradapters! Eine Beschädigung wäre nicht auszuschließen! Zum Abgleich sind folgende Schritte nötig:

1. mc-EPROM-Brenner in Modus B schalten.
2. Adapterplatine in den EPROM-Sockel einsetzen.
3. Schließen Sie ein Voltmeter zwischen Anschluß 20 und Anschluß 7 des Einchip-Mikrocomputer-Programmiersokkels X1 an. Der Meßbereichsendwert des Meßgerätes sollte größer oder gleich 20 V sein.

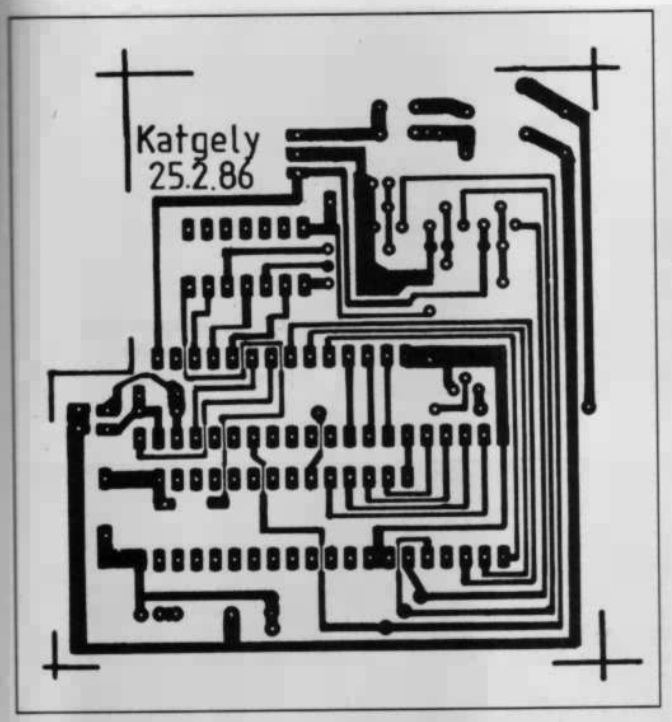


Bild 5. Platinen-Layout für den Programmieradapter

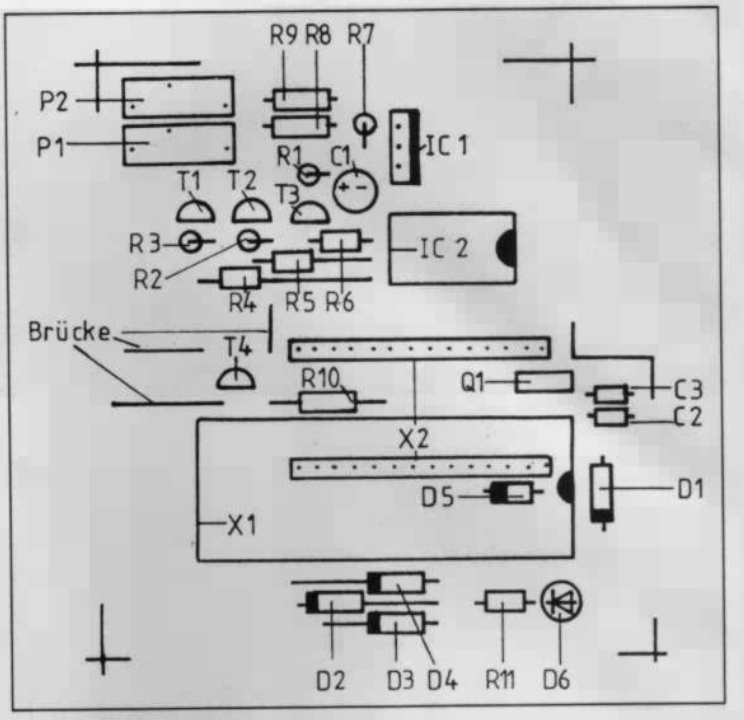


Bild 6. Bestückungsplan des Programmieradapters

4. Starten Sie das geänderte Turbo-Pascal-Programm.
5. Geben Sie ein: T 60
6. Geben Sie ein: F 0 B 2000 M FF
7. Geben Sie ein: P 0
8. Stellen Sie, während der Programmiervorgang läuft, mit P1 eine Spannung von 18 V ein. Sollte die Programmierung abgeschlossen sein, bevor Sie die korrekte Spannung eingestellt haben, wiederholen sie einfach die Schritte 7 und 8.
9. Nach abgeschlossener Programmierung und korrekter Einstellung geben Sie ein: T 62
10. Geben Sie ein: P 0
11. Stellen Sie mit P2 eine Spannung von 12 V ein. Sollte die Programmierung abgeschlossen sein, bevor Sie die korrekte Spannung eingestellt haben, wiederholen Sie einfach die Schritte 10 und 11.

Damit ist der Abgleich der Hardware abgeschlossen.

Die Treibersoftware

Zur Programmierung von MCS48-Einchip-Mikrocomputern muß sich der mc-EPROM-Brenner in jedem Fall in Modus B befinden. Dadurch beziehen sich alle hier beschriebenen Änderungen des Programmes auf die Original-Programmversion aus der mc 1/87 für Modus B. Die Besitzer des mc-EPROM-Brenners müssen lediglich die alten Programmteile gegen die geänderten Programmteile austauschen, die neu hinzugekommenen Funktionen und Prozeduren einfügen und die Pascal-Quelldatei neu compilieren.
In folgenden Programmteilen wurden Änderungen durchgeführt:

TYPE_CMD	READ_CMD
TEST_CMD	IDENTIFY_CMD
PROGRAM_CMD	VERIFY_CMD

Ferner wurden Änderungen in den zum Hauptprogramm MAIN gehörigen Deklarationen durchgeführt.
Für die Besitzer anderer EPROM-Programmiergeräte wird es schwieriger werden, die entsprechenden Änderungen durchzuführen. Sie sind gezwungen, die entsprechenden Routinen selbst zu schreiben.
Zur Anpassung der Software an andere EPROM-Programmiergeräte sollten die entsprechenden Datenblätter der Firma Intel zu Rate gezogen werden.

Die Software im Detail

Folgende Änderungen wurden in den Deklarationen des Hauptprogramms vorgenommen:

1. Die Felder RD, VER, PROG, PINH des Records EPROM_DATA wurden von array [3..4] of byte auf array [2...4] of byte geändert.
2. Der Typendeklarationsteil des Hauptprogramms wurde um die Deklaration des Typs MODI = (MCS48, EPROMS); erweitert.
3. Der Variablendeklarationsteil des Hauptprogramms wurde um die Deklaration der Variablen MODE : MODI;
   PRIMARAY : boolean; erweitert.
   Hierbei gibt die Variable MODE an, ob EPROMs oder Einchip-Mikrocomputer bearbeitet werden sollen. Die Variable PRIMARY gibt an, ob die für die MCS48-Einchip-Mikrocomputer erforderliche Initialisierung bereits durchgeführt wurde (PRIMARY = false) oder ob diese Initialisierung noch durchgeführt werden muß (PRIMARY = true).
4. Die Konstante EPROM_TYPES wurde von 18 auf 21 erhöht.
5. Das Array KNOWN_EPROMS wurde gemäß *Bild 7* geändert und erweitert. Dabei ist zu beachten, daß bei allen EPROM-Typen dieses Arrays die Felder RD, VER, PROG, PINH um $00 erweitert werden müssen. In Bild 7 ist diese Änderung am Beispiel des EPROMs 2764 gezeigt. Die dort vorgenommenen Änderungen müssen analog bei allen anderen EPROMs vorgenommen werden. Neu zu der Deklaration des Array KNOWN_EPROMs hinzugekommen sind die Elemente 8048, 8748 und 8749. Eine komplette Auflistung der Typenbeschreibungen aller bekannten EPROMs und Einchip-Mikrocomputer im neuen Format zeigt *Bild 8*.

```
(TYP  : '2764          '; LEN  : $2000      ;   TIME : 50          ;
 TNR  : '30'           ; VPP  : '21 V  '    ;   PGMD : 'N'         ;
                         RD   : ($C0,$10)   ;   VER  : ($40,$50)   ;
                         PROG : ($00,$56)   ;   PINH : ($40,$13) ),

Alte EPROM-Typenbeschreibung.

(TYP : '2764           '; LEN : $2000       ; TIME: 50             ;
 TNR : '30'            ; VPP : '21 V  '     ; PGMD: 'N'            ;
                         RD  : ($00,$C0,$10); VER : ($00,$40,$50)  ;
                         PROG: ($00,$00,$56); PINH: ($00,$40,$13)  ),

Neue EPROM-Typenbeschreibung.
```

Bild 7. Gegenüberstellung der alten und neuen Typenbeschreibung

```
const  EPROM_TYPES = 21;  (* Anzahl der bekannten EPROMS u. SINGLECHIPS *)
       KNOWN_EPROMS : array [1..EPROM_TYPES] of EPROM_DATA =

((TYP  : '2716         '; LEN  : $0800      ; TIME : 00            ;
 TNR  : '10'           ; VPP  : '     '     ; PGMD : 'Z'           ;
                         RD   : ($00,$28,$00); VER  : ($00,$00,$00) ;
                         PROG : ($00,$00,$00); PINH : ($00,$00,$00) ),

(TYP  : '2732          '; LEN  : $1000      ; TIME : 50            ;
 TNR  : '20'           ; VPP  : '25 V  '    ; PGMD : 'O'           ;
                         RD   : ($00,$20,$00); VER  : ($00,$20,$18) ;
                         PROG : ($00,$20,$3C); PINH : ($00,$20,$39) ),

(TYP  : '2732A         '; LEN  : $1000      ; TIME : 50            ;
 TNR  : '21'           ; VPP  : '21 V  '    ; PGMD : 'O'           ;
                         RD   : ($00,$20,$00); VER  : ($00,$20,$00) ;
                         PROG : ($00,$20,$24); PINH : ($00,$20,$21) ),

(TYP  : '2764          '; LEN  : $2000      ; TIME : 50            ;
 TNR  : '30'           ; VPP  : '21 V  '    ; PGMD : 'N'           ;
                         RD   : ($00,$C0,$10); VER  : ($00,$40,$50) ;
                         PROG : ($00,$00,$56); PINH : ($00,$40,$13) ),

(TYP  : '2764          '; LEN  : $2000      ; TIME : 1             ;
 TNR  : '31'           ; VPP  : '12,5 V'    ; PGMD : 'A'           ;
                         RD   : ($00,$C0,$10); VER  : ($00,$40,$C8) ;
                         PROG : ($00,$00,$CE); PINH : ($00,$40,$8B) ),

(TYP  : '2764          '; LEN  : $2000      ; TIME : 1             ;
 TNR  : '32'           ; VPP  : '21 V  '    ; PGMD : 'B'           ;
                         RD   : ($00,$C0,$10); VER  : ($00,$40,$48) ;
                         PROG : ($00,$00,$4E); PINH : ($00,$40,$0B) ),

(TYP  : '2764          '; LEN  : $2000      ; TIME : 1             ;
 TNR  : '33'           ; VPP  : '12,5 V'    ; PGMD : 'C'           ;
                         RD   : ($00,$C0,$10); VER  : ($00,$40,$C8) ;
                         PROG : ($00,$00,$CE); PINH : ($00,$40,$8B) ),

(TYP  : '2764          '; LEN  : $2000      ; TIME : 1             ;
 TNR  : '34'           ; VPP  : '21 V  '    ; PGMD : 'C'           ;
                         RD   : ($00,$C0,$10); VER  : ($00,$40,$48) ;
                         PROG : ($00,$00,$4E); PINH : ($00,$40,$08) ),

(TYP  : '27128         '; LEN  : $4000      ; TIME : 50            ;
 TNR  : '40'           ; VPP  : '21 V  '    ; PGMD : 'N'           ;
                         RD   : ($00,$C0,$10); VER  : ($00,$40,$50) ;
                         PROG : ($00,$00,$56); PINH : ($00,$40,$13) ),

(TYP  : '27128         '; LEN  : $4000      ; TIME : 1             ;
 TNR  : '41'           ; VPP  : '12,5 V'    ; PGMD : 'A'           ;
                         RD   : ($00,$C0,$10); VER  : ($00,$40,$C8) ;
                         PROG : ($00,$00,$CE); PINH : ($00,$40,$8B) ),

(TYP  : '27128         '; LEN  : $4000      ; TIME : 1             ;
 TNR  : '42'           ; VPP  : '21 V  '    ; PGMD : 'B'           ;
                         RD   : ($00,$C0,$10); VER  : ($00,$40,$48) ;
                         PROG : ($00,$00,$4E); PINH : ($00,$40,$0B) ),

(TYP  : '27128         '; LEN  : $4000      ; TIME : 1             ;
 TNR  : '43'           ; VPP  : '12,5 V'    ; PGMD : 'C'           ;
                         RD   : ($00,$C0,$10); VER  : ($00,$40,$C8) ;
                         PROG : ($00,$00,$CE); PINH : ($00,$40,$8B) ),

(TYP  : '27128         '; LEN  : $4000      ; TIME : 1             ;
 TNR  : '44'           ; VPP  : '21 V  '    ; PGMD : 'C'           ;
                         RD   : ($00,$C0,$10); VER  : ($00,$40,$48) ;
                         PROG : ($00,$00,$4E); PINH : ($00,$40,$0B) ),

(TYP  : '27256         '; LEN  : $8000      ; TIME : 50            ;
 TNR  : '50'           ; VPP  : '21 V  '    ; PGMD : 'N'           ;
                         RD   : ($00,$80,$10); VER  : ($00,$00,$51) ;
                         PROG : ($00,$00,$56); PINH : ($00,$00,$17) ),

(TYP  : '27256         '; LEN  : $8000      ; TIME : 1             ;
 TNR  : '51'           ; VPP  : '12,5 V'    ; PGMD : 'A'           ;
                         RD   : ($00,$80,$10); VER  : ($00,$00,$C9) ;
                         PROG : ($00,$00,$CE); PINH : ($00,$00,$8F) ),

(TYP  : '27256         '; LEN  : $8000      ; TIME : 1             ;
 TNR  : '52'           ; VPP  : '21 V  '    ; PGMD : 'B'           ;
                         RD   : ($00,$80,$10); VER  : ($00,$00,$49) ;
                         PROG : ($00,$00,$4E); PINH : ($00,$00,$0F) ),

(TYP  : '27256         '; LEN  : $8000      ; TIME : 1             ;
 TNR  : '53'           ; VPP  : '12,5 V'    ; PGMD : 'C'           ;
                         RD   : ($00,$80,$10); VER  : ($00,$00,$C9) ;
                         PROG : ($00,$00,$CE); PINH : ($00,$00,$8F) ),

(TYP  : '27256         '; LEN  : $8000      ; TIME : 1             ;
 TNR  : '54'           ; VPP  : '21 V  '    ; PGMD : 'C'           ;
                         RD   : ($00,$80,$10); VER  : ($00,$00,$49) ;
                         PROG : ($00,$00,$4E); PINH : ($00,$00,$0F) ),

(TYP  : '8048          '; LEN  : $0400      ; TIME : 00            ;
 TNR  : '60'           ; VPP  : '     '     ; PGMD : 'R'           ;
                         RD   : ($80,$81,$4D); VER  : ($98,$81,$4D) ;
                         PROG : ($F0,$81,$4D); PINH : ($80,$81,$4D) ),

(TYP  : '8748          '; LEN  : $0400      ; TIME : 50            ;
 TNR  : '61'           ; VPP  : '21 V  '    ; PGMD : 'N'           ;
                         RD   : ($80,$81,$4C); VER  : ($98,$81,$4C) ;
                         PROG : ($F0,$81,$4C); PINH : ($80,$81,$4C) ),

(TYP  : '8749          '; LEN  : $0800      ; TIME : 50            ;
 TNR  : '62'           ; VPP  : '21 V  '    ; PGMD : 'N'           ;
                         RD   : ($80,$81,$4C); VER  : ($98,$81,$4C) ;
                         PROG : ($F0,$81,$4C); PINH : ($80,$81,$4C) ));
```

Bild 8. Typenbeschreibung der bekannten Bausteintypen

Änderungen in TYPE_CMD:

- Deklaration der Variablen TYPNUMBER im Typendeklarationsteil

```
TYPNUMBER : integer;
```

- Einfügen der Anweisungen

```
val (KNOWN_EPROMS [W].TNR, TYPNUMBER, E1);
case TYPNUMBER of
  60..69 : MODE := MCS48;
  else MODE := EPROMS;
end; (* case *)
```

vor die Anweisung

```
if EPROM.LEN = $8000 ....
```

Damit sind die Änderungen an der Prozedur TYPE_CMD abgeschlossen.

Änderungen in TEST_CMD:

- Deklaration der Variablen EMPTY im Variablendeklarationsteil.

```
EMPTY : byte;
```

- Einfügen der Anweisung

```
if MODE = MCS48 then EMPTY := $00;
     else EMPTY := $FF;
```

vor die DISP(15,11.....) Anweisung.

- Ersetzen der Anweisung

```
DATA_EPROM := GET_EPROM (I);
```

durch die Anweisung

```
case MODE of
  MCS48 : DATA_EPROM := GET_MCS48 (I);
  EPROMS: DATA_EPROM := GET_EPROM (I);
end; (* case *)
```

Weitere Änderungen müssen im TEST_CMD nicht durchgeführt werden.

Änderungen in VERIFY_CMD:

- Ersetzen der Anweisung

```
DATA_EPROM := GET_EPROM (I);
```

durch die Anweisung

```
case MODE of
  MCS48 : DATA_EPROM := GET_MCS48 (I);
  EPROMS: DATA_EPROM := GET_EPROM (I);
end; (* case *)
```

Damit sind alle nötigen Änderungen an VERIFY_CMD durchgeführt.

Änderungen in PROGRAM_CMD:

- Ersetzen der Anweisung

```
DATA_EPROM := PUT_EPROM (I, DATA_RAM);
```

Bild 9. Änderungen an der Original-Software

durch die Anweisung

```
case MODE of
  MCS48 : DATA_EPROM := PUT_MCS48 (I, DATA_RAM);
  EPROMS: DATA_EPROM := PUT_EPROM (I, DATA_RAM);
end;'(* case *)
```

Diese Änderunge bezieht sich nur auf den 'N'-Fall (Normale Programmierung) der schon im PROGRAM_CMD vorhandenen case-Anweisung.

- Erweitern der übergeordneten case-Anweisung um den Fall 'R' (ROM - Version. Sie kann nur gelesen , aber nicht programmiert werden !).

```
'R': begin
       clrscr; write(#07); gotoxy (18,10);
       writeln('*********************************');
       gotoxy (18,11);
       writeln('*       ROM-VERSION !!          *');
       gotoxy (18,12);
       writeln('*  Programmierung unmöglich ! *');
       gotoxy (18,13);
       writeln('*********************************');
       delay (1000);
       goto END1; (* Fehler ! Beende PROGRAM_CMD ! *)
```

- Einfügen der Anweisung

```
PRIMARY := true;
```

hinter das end; der übergeordneten case-Anweisung, unmittelbar vor der Anweisung gatoxy (1,19);.

- Ersetzen der Anweisung

```
DATA_EPROM := GET_EPROM (I);
```

im letzten Teil des PROGRAM_CMD's (12. Zeile von unten ab END1) durch die Anweisung

```
case MODE of
  MCS48 : DATA_EPROM := GET_MCS48 (I);
  EPROMS: DATA_EPROM := GET_EPROM (I);
end; (* case *)
```

Weitere Änderungen müssen im PROGRAM_CMD nicht vorgenommen werden.

Änderungen im IDENTIFY_CMD:

- Einfügen der Anweisung

```
case MODE of
  EPROMS: begin
```

als erste Anweisung in der Prozedur IDENTIFY_CMD.

- Einfügen der Zeilen

```
end;
MCS48 : begin
          clrscr; write(#07); gotoxy (18,10);
          writeln ('*********************************');
          gotoxy (18,11);
          writeln ( *      MCS48-Einchip-       * );
          gotoxy (18,12);
          writeln ( *  Mikrocomputer können nicht  *');
          gotoxy (18,13);
          writeln ( *    identifiziert werden !    *');
          gotoxy (18,14);
          writeln ( *********************************');
          delay(1000);
        end;
end; (* case *)
```

vor das die Prozedur IDENTIFY_CMD abschließende end; .

```
procedure INIT_MCS48;
begin
  OUT_BYTE [1] := $00;
  OUT_BYTE [2] := ((EPROM.RD [2] or $08) and $7f);
  OUT_BYTE [3] := EPROM.RD [3];
  OUT_BYTE [4] := EPROM.RD [4];
  PUT_DATA;
  OUT_BYTE [2] := OUT_BYTE [2] or $80;  (* EA auf 18 V *)
  PUT_DATA;
  PRIMARY := false;
end;

function GET_MCS48 (ADR1 : real) : byte;
const   LATCH_READ = $18;
var     ADR : integer;
begin
  ADR := trunc (ADR1);
  if PRIMARY then INIT_MCS48;   (* Wurde schon initialisiert ? *)
  OUT_BYTE [1] := lo (ADR);
  OUT_BYTE [2] := hi (ADR) or EPROM.RD [2];
  OUT_BYTE [3] := EPROM.RD [3];
  OUT_BYTE [4] := EPROM.RD [4];
  PUT_DATA;
  OUT_BYTE [2] := OUT_BYTE [2] or LATCH_READ;
  PUT_DATA;
  OUT_BYTE [4] := OUT_BYTE [4] and $FB;
  PUT_DATA;
  GET_MCS48 := GET_DATA;
end;

function PUT_MCS48 (ADR1 : real; Data : byte): byte;
const    LATCH_WRITE = $10;
var      ADR : integer;
begin
  ADR := trunc (ADR1);
  if PRIMARY then INIT_MCS48; (* Wurde schon initialisiert ? *)
  OUT_BYTE [1] := lo (ADR);
  OUT_BYTE [2] := hi (ADR) or EPROM.PINH [2];
  OUT_BYTE [3] := EPROM.PINH [3];
  OUT_BYTE [4] := EPROM.PINH [4];
  PUT_DATA;
  OUT_BYTE [2] := OUT_BYTE [2] or LATCH_WRITE;
  PUT_DATA; (* A0 bis A7 im Singelchip zwischenspeichern *)
  OUT_BYTE [1] := DATA;
  OUT_BYTE [2] := hi (ADR) or EPROM.PROG [2];
  OUT_BYTE [3] := EPROM.PROG [3];
  OUT_BYTE [4] := EPROM.PROG [4];
  PUT_DATA; (* Datenbyte in Singlechip brennen *)
  delay (EPROM.TIME);
  OUT_BYTE [2] := EPROM.PINH [2] or LATCH_WRITE;(* Prog.puls aus *)
  PUT_DATA;
  OUT_BYTE [2] := OUT_BYTE [2] or $08; (* TO auf H *)
  OUT_BYTE [4] := OUT_BYTE [4] and $FB;
  PUT_DATA;
  PUT_MCS48 := VERIFY_MCS48 (ADR);
end;
```

Bild 10. Zusätzliche Prozeduren und Funktionen

```
IC 1            LM 317
IC 2            7406
T1..T3          BC 327
T4              BC 546
D1, D2          1 N 4002
D3..D5          1 N 4148
D6              LED 5mm rot
C1              TANTALELKO 10 uF, 35 V
C2,C3           20 pF, keramisch

R1..R3          3,3 K, 1/4 Watt
R4..R6          10  K, 1/4 Watt
R7              220 R, 1/4 Watt
R8              2,7 K, 1/4 Watt
R9              5,6 K, 1/4 Watt
R10             10  K, 1/4 Watt
R11             330 R, 1/4 Watt
P1              5 K, Spindeltrimmer
P2              5 K, Spindeltrimmer

Q1              Quarz 3 MHz

X1              Textoolsockel 40 polig
X2              Pfostensteckerleisten 2,54 mm
                einfach, 2x14 Kontakte vergoldet

 1 x 14poliger DIL-Sockel
 1 x Platine gemäß Layout Bild 8
```

Bild 11. Mit diesen Bauteilen wird der Programmieradapter aufgebaut

6. Die Deklaration der Konstanten EMPTY entfällt. An ihre Stelle tritt eine in der Prozedur TEST_CMD deklarierte Variable EMPTY. Diese Änderung wurde dadurch nötig, daß der Programmspeicher der MCS48-Einchip-Mikrocomputer im unprogrammierten Zustand $00 enthält, wogegen unprogrammierte EPROMs $FF enthalten. Um eine einwandfreie Erkennung gelöschter Bausteine in jeder Betriebsart zu gewährleisten, muß je nach Betriebsmode (MCS48, EPROMs) ein anderes EMPTY-Muster verwendet werden.

Bild 9 zeigt die restlichen Änderungen an der Software. Die Prozedur- und Funktionsdeklarationen wurden um die Prozedur INIT_MCS48 und die Funktionen GET_MCS48, PUT_MCS48 und VERIFY_MCS48 erweitert. Das Listing der neuhinzugefügten Funktionen kann *Bild 10* entnommen werden. Beim Einbau der neuen Prozeduren bzw. Funktionen sind die Prozedur INIT_MCS48 und die Funktion GET_MCS48 vor die Funktion GET_EPROM und die Funktionen VERIFY_MCS48 und PUT_MCS48 vor die Funktion PUT_EPROM in den Quellcode einzubinden.

Inbetriebnahme

Sind alle Änderungen im Programm vorgenommen und ist das geänderte Programm compiliert, die Hardware aufgebaut und abgeglichen, dann steht einer Inbetriebnahme nichts mehr im Wege. Um unnötigen Ärger zu vermeiden, empfiehlt es sich jedoch, nicht sofort einen Einchip-Mikrocomputer in den Programmieradapter einzusetzen und „loszuprogrammieren". Vielmehr ist es sinnvoll, den EPROM-Brenner erst einmal ohne eingesetzten Baustein zu betreiben und verschiedene Spannungspegel zu überprüfen. So sollten Sie überprüfen, ob die an Anschluß 7 anliegende Spannung beim Betrieb nicht über 12 V (für 8048/8049) bzw. 18 V (für 8748/8749) liegt. Ebenso empfiehlt sich eine Kontrolle der Betriebsspannung am Anschluß 40. Diese Spannung sollte nicht über 5,5 V liegen.

Literatur

[1] *Seng, Peter:* mc-EPROM-Brenner, mc 1987, Heft 1, Seite 46...58.

[2] *Plate, Jürgen:* Sichere Übertragung. mc 1987, Heft 3, Seite 104..108.

[3] *Pelka, Horst:* Der Ein-Chip-Mikrocomputer, Franzis-Verlag.

[4] Microcontroller Handbook, Fa. Intel.

Eckhard Schadwinkel, Heinrich Rescher

Videotext in den Mikrocomputer

Teil 2

Wir verwenden, wie viele Fernsehgerätehersteller auch, Videotext-Chips der Firma VALVO. Anstelle des für die Decodersteuerung von VALVO vorgeschlagenen Mikrocontrollers haben wir jedoch die Schaltung mit einem anderen Mikroprozessor versehen. Bei uns arbeitet ein 8031.

Der Videotext-Decoder muß zwei grundlegende Aufgaben erfüllen. Einerseits blendet er die eigentliche Videotext-Information aus dem Bildsignal aus und erzeugt die nötigen Synchronsignale. Andererseits muß er diese Daten nach verschiedenen wählbaren Kriterien aufarbeiten und darstellen.

Bei dem von uns verwendeten Videotext-Decoder handelt es sich um einen Decoder der zweiten Generation. Er unterscheidet zwischen beiden Aufgabenbereichen und erlaubt auf sehr ökonomische Weise eine hohe Anwendungsflexibilität: Die festen Funktionen werden durch zugeordnete Schaltungen in den ICs ausgeführt, die variablen Funktionen laufen unter Programmsteuerung eines angeschlossenen Controllers ab.

Der eigentliche Decoder besteht aus nur zwei ICs (plus Speicherbausteinen). Der Videoprozessor SAA 5230 (VIP 2 = Video Input Processor, siehe *Bild 1*) trennt die Videotext-Daten aus dem Videosignal ab. Erwähnenswert ist in diesem Zusammenhang, daß die Videoprozessoren selbsttätig alle Zeilen in den Austastlücken erkennen, die Videotext-Daten enthalten. Dadurch ist sichergestellt, daß auch dann, wenn später weitere Zeilen für Videotext genutzt werden, die Decoder voll funktionsfähig bleiben. Nach dem Abtrennen bereitet der VIP 2 die Daten auf und gibt sie an die Schaltung SAA5240 weiter. Dieser Schaltkreis ist der sogenannte EURO CCT (=European Computer-Controlled Teletext chip), die Bezeichnung EURO weist auf den 625 Zeilen-Betrieb hin). *Bild 2* zeigt das Prinzipschaltbild des EURO CCT. Er erfüllt nicht nur die Forderungen augenblicklicher Videotext-Übertragungen, er ist auch mit zukunftsweisenden Möglichkeiten ausgestattet:

– durch spezielle Steuerbits in der Seitenüberschrift automatisches Umschalten des Zeichensatzes auf eine von drei Sprachgruppen

– auch Datenempfang in allen Zeilen eines Vollbildes (= full channel mode) mit schnellstem Seitenzugriff möglich

– Darstellung von Status-Informationen in einer zusätzlichen, vom Decoder erzeugten Textzeile oberhalb oder unterhalb des Haupttextes

– Zusatzinformationen in 24 zusätzlichen virtuellen Textreihen, sogenannten ghost rows

– gleichzeitiger Suchvorgang von bis zu vier Seiten im Videotext-Signal

Es würde den Umfang des Beitrags sprengen, wollte man alle Funktionsblöcke be-

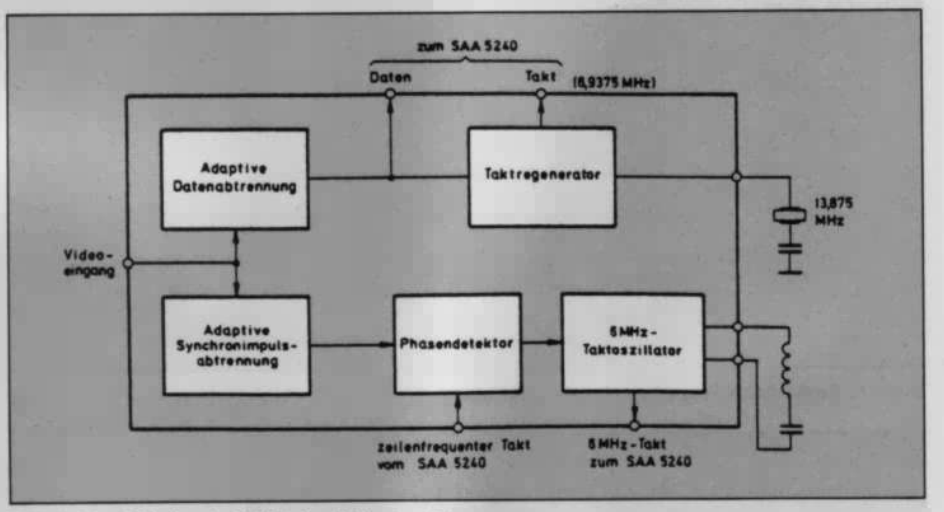


Bild 1. Das Prinzipschaltbild des Videotext-Prozessors

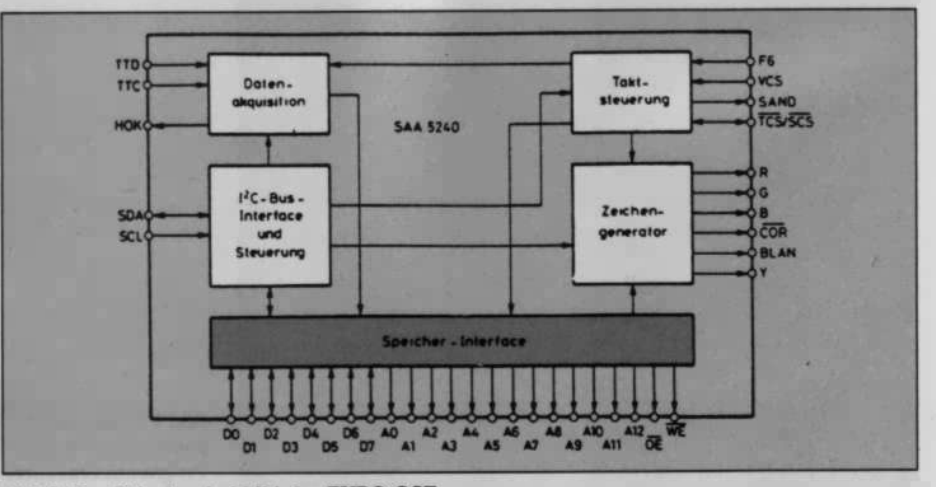


Bild 2. Das Prinzipschaltbild des EURO-CCT

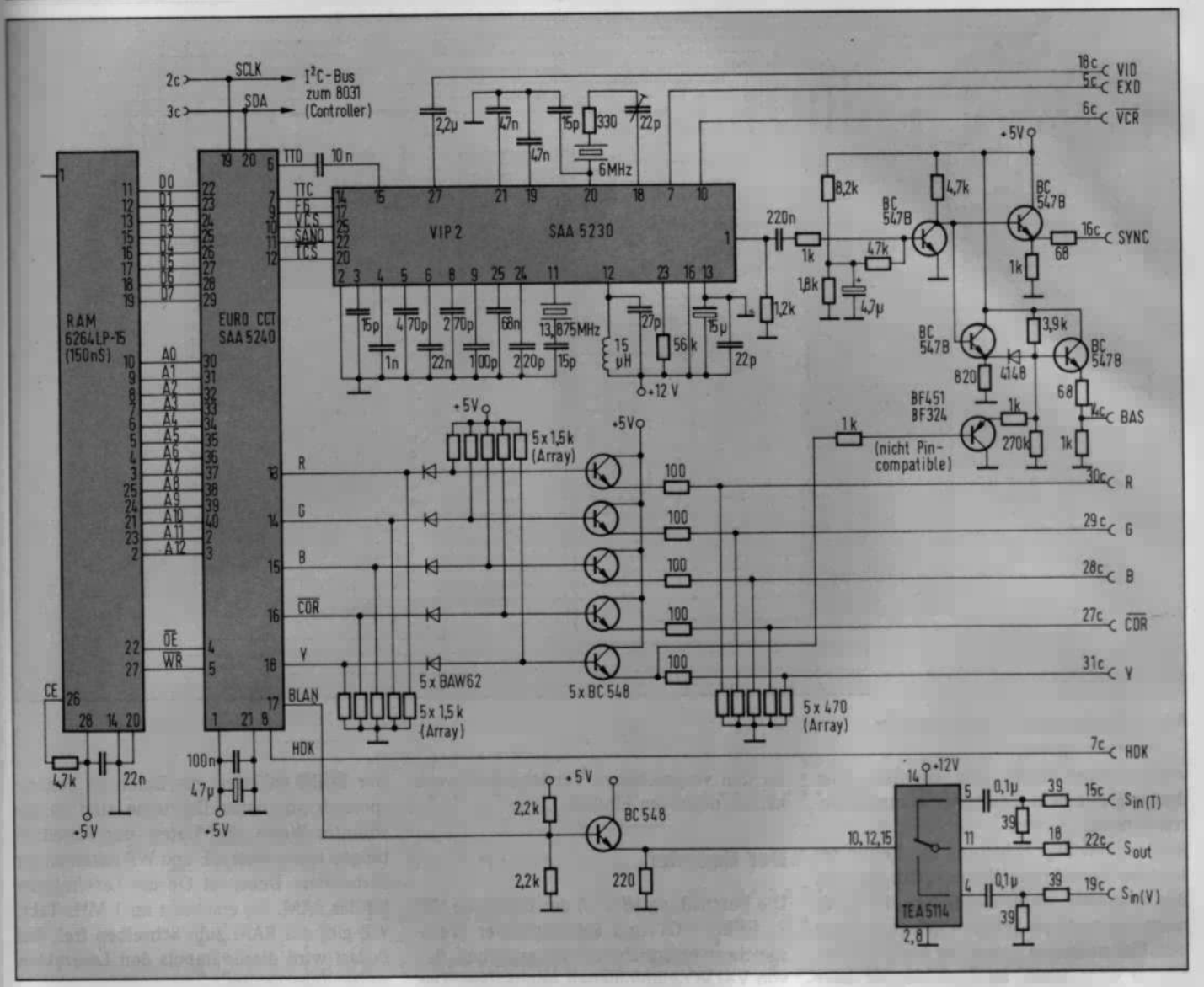


Bild 3. Das Schaltbild des Decoders

schreiben. Deshalb sollen nur die für die Rechnerkopplung wichtigsten Blöcke, erstens der Zeichengenerator und zweitens der Block I²C-Bus-Interface und Steuerung kurz erläutert werden. Beide Module befinden sich im EURO CCT.

Der Zeichengenerator

Die wichtigsten Leistungsmerkmale des Zeichengenerators sind in *Tabelle 2* festgehalten (in der vorherigen Ausgabe von mc). Dazu einige Erläuterungen: Die 12x10 Punktmatrix liefert eine feinere Zeichenstruktur als bei den Decodern der ersten Generation und damit eine bessere Zeichenlesbarkeit.

Die Austastsignale für den Fernsehbildschirm werden über den Anschluß BLAN ausgegeben. Es sind hierüber Rasterpunkt-, Umfeld- und Vollbildaustastung möglich. Steuerzeichen für Blinken, Box, doppelte Höhe, verdeckte Anzeige, Farbe und Grafik werden ausgewertet.

I²C-Bus-Interface und Steuerung

Dieser Funktionsblock enthält die Elemente zur Steuerung der verschiedenen EURO CCT-Funktionen. Die Steuerung erfolgt entweder direkt unter Verwendung der Moderegister-Bits (Diese Möglichkeit nutzt der Microcontroller. Über den WI-Befehl können die elf Register direkt beschrieben und damit besondere Funktionen gewählt werden) oder indirekt über die Speicherschaltungen. Der I²C-Bus-Slave-Sender/Empfänger nimmt die Steuerbefehle über die Anschlüsse SDA und SCL (serielle Daten beziehungsweise serieller Takt) entgegen. Die wichtigsten Merkmale zeigt Tabelle 3 (in mc, Ausgabe 3).

Der I²C-Bus

I²C steht für IIC, was Inter IC bedeutet. Der I²C-Bus ist ein aus nur zwei Verbindungen bestehender, bidirektioneller Multimaster-Bus mit serieller Datenübertragung. Seine Datenleitung SDA und seine Taktleitung SCL werden extern über Widerstände an U_{DD} gelegt. Durch die Wahl eines seriellen Datentransfers zwischen den einzelnen Bauelementen vereinfacht sich das Platinen-Layout und die Platinen werden kleiner. Außerdem verringert sich die Anzahl von Steckleistenkontakte. Daher ist der I²C-Bus besonders zur Verknüpfung von elektronischen Bausteinen in einem gemeinsamen Gehäuse geeignet. Es existieren mittlerweile eine Reihe von peripheren Schaltungen mit serieller Schnittstelle (LED-, LCD-Ansteuerungen, Uhren-Timer-Schaltungen, RAMs, nichtflüchtige Speicher und

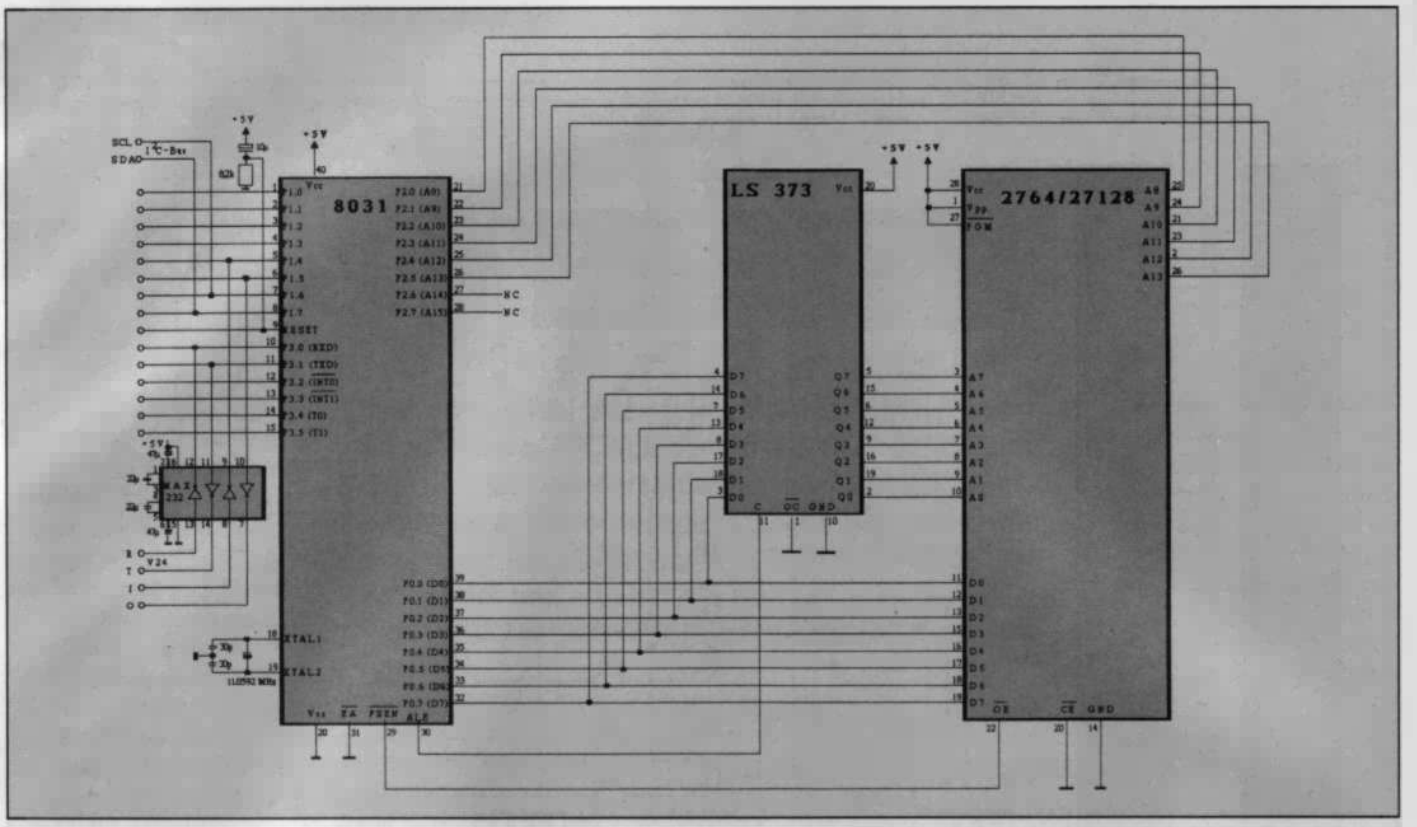


Bild 4. Das Schaltbild des Controllers

verschiedenes mehr), die ebenfalls über den Bus mit dem Controller kommunizieren können.
Jede sendewillige Schaltung kann ohne besondere Vorkehrung immer dann auf den Bus zugreifen, wenn dieser frei, das heißt nicht von anderen Sendern belegt ist. Jede den Bus nutzende Schaltung kann sowohl Master oder Slave, als auch Sender oder Empfänger sein. Es sind vier Betriebsarten möglich:

- Master-Sender
- Master-Empfänger
- Slave-Sender
- Slave-Empfänger

Der Master unterscheidet sich vom Slave dadurch, daß er (der Master) unabhängig davon, ob er Sender oder Empfänger ist, den Takt erzeugt und damit die Frequenz der seriellen Datenübertragung bestimmt. Die Taktfrequenz wird über Software festgelegt. Bei dem von VALVO benutzten Prozessor Typ MAB 84xx mit einem Systemtakt von 4,43 MHz läßt sie sich durch einen in die serielle Schnittstelle integrierten Baudrate-Generator zwischen 720 Hz und 110 kHz einstellen. Bei der von uns gewählten Lösung, die die I^2C-Bus-Steuerung softwaremäßig realisiert, kann kein Multimasterbetrieb stattfinden. Dies hat jedoch für den vorgesehenen Verwendungszweck keinen negativen Einfluß.

Der Decoder

Die Beschaltung (*Bild 3*) der Bausteine VIP 2, EURO CCT und Seitenspeicher (Textspeicher) entspricht im wesentlichen der von VALVO empfohlenen Applikation. Das Videoeingangssignal (VID) wird über einen Kondensator dem Baustein VIP 2 zugeführt. Die Leitungen TTD, TTC, F6, VCS, SAND und $\overline{TCS}$ führen vom VIP2 zum EURO CCT. Sie haben folgende Bedeutung:

TTD =>
Videotext-Daten, dies sind die aus den Bildschirmzeilen gewonnen VT-Daten.
TTC =>
Videotext-Datentakt, dies ist eine Frequenz von 6,9375 MHz, die der VIP 2 bereitstellt.
F6 =>
6 MHz – Systemtakt
VCS =>
vom Videoeingangssignal (FBAS) abgetrenntes Zeilensync-Signal
SAND =>
Sandcastle-Impuls vom EURO CCT für den Phasenvergleich
$\overline{TCS}$ =>
Text Composite Sync – Impuls vom EURO CCT, zur Zeilensynchronisation bei Textwiedergabe mit/ohne Zeilensprung.

Der EURO CCT legt die Daten im Seitenspeicher ab. Beide Bausteine sind in gewohnter Weise über Daten- und Adreßleitungen sowie über $\overline{OE}$ und $\overline{WR}$ miteinander verbunden. Dabei ist $\overline{OE}$ die Lesefreigabe für das RAM. Sie erscheint im 1 MHz-Takt. $\overline{WE}$ gibt das RAM zum Schreiben frei. Bei Bedarf wird dieser Impuls den Lesezyklen zwischengeschaltet.
Werden Daten aus dem RAM ausgelesen, gelangen sie nach Aufarbeitung zu den Anschlüssen R, G, B, $\overline{COR}$, Y und BLAN des EURO CCT. Sie haben folgende Bedeutung:

R, G und B =>
dies sind Open-Drainausgänge (für Rot, Grün und Blau) zur Ansteuerung der Fernsehgerät-Schnittstelle. Die Signale beinhalten sowohl das Bildsignal als auch den Hintergrund.
$\overline{COR}$ =>
dies ist ein Open-Drainausgang zur geschalteten Kontrastreduzierung des Fernsehbildes bei Misch- und Einblendbetrieb (wird in dieser Schaltung nicht genutzt).
Y =>
Y-Signal mit 100%-Schwarzpegel ohne Sync-Signale und ohne Blinkfunktion zum Beispiel zum Anschluß eines Druckers.
BLAN =>
blanking, Austastungsignal für die Video-Austastung im Fernsehgerät bei Misch- und Einblendbetrieb.

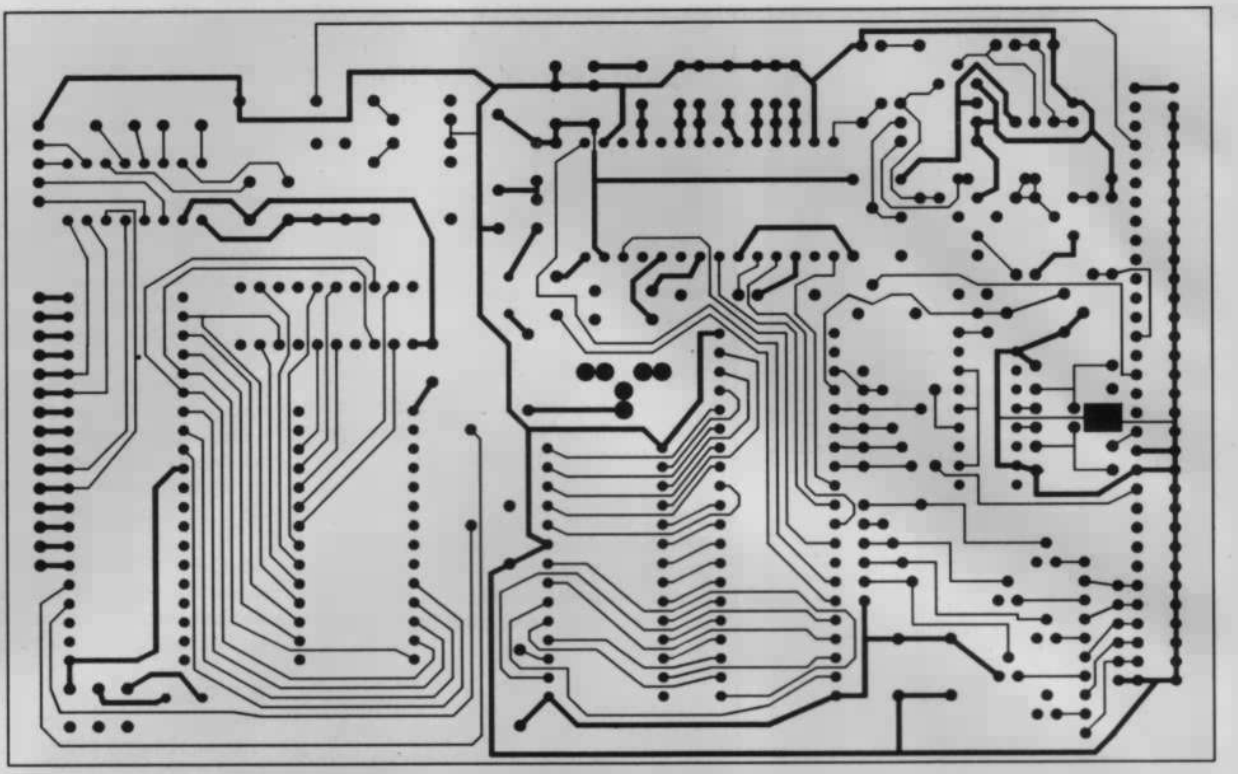


Bild 5.
Das Layout der Lötseite

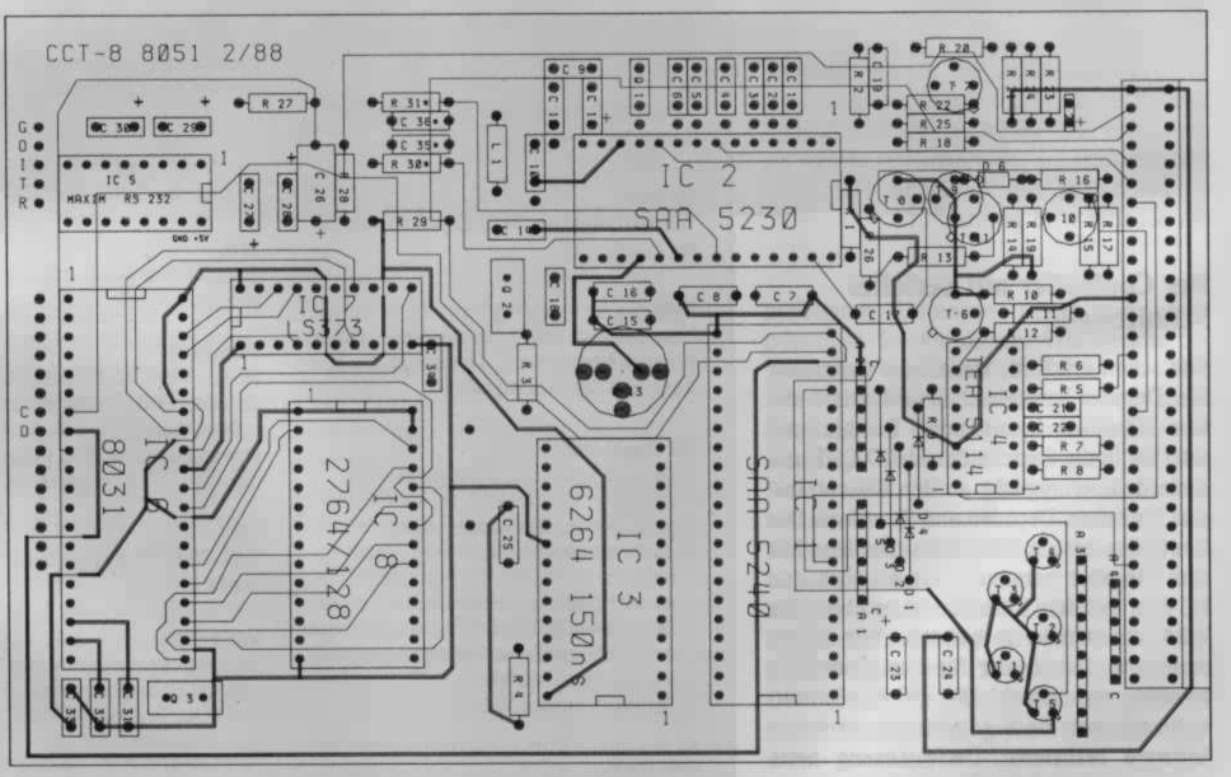


Bild 6.
Die Bestückungsseite

Alle Signale werden über Emitterfolger zur Steckleiste geführt.
Pin 19 und 20 des EURO CCT stellen die Leitungen des I^2C-Bus dar. Sie führen zum Controller 8031.
Zwei benötigte Signale müssen besonders aufbereitet werden:
Aus dem Signal an Pin 1 vom VIP 2 wird über einen Transistor BC547 B das SYNC-Signal gewonnen. Die nachfolgende Transistorstufe paßt den Pegel an.
Das Y-Signal wird hinter dem oben beschriebenen Emitterfolger (vor dem 100-Ohm-Längswiderstand) abgegriffen und einem weiteren Emitterfolger zugeführt. Dieser Schaltungsteil besteht wegen der Pegelverhältnisse aus einem pnp-Transistor. Am Emitter wird das oben beschriebene SYNC-Signal hinzugemischt; es entsteht das sogenannte BAS-Signal. Eine nachfolgende Transistorstufe dient zur Pegelanpassung. Dieses BAS-Signal liefert unter anderem das schwarz/weiße Textsignal, das über einen Videoschalter dem, vom Empfangsteil gelieferten, FBAS-Signal zugemischt wird. Das BLAN-Signal dient als Schaltimpuls. Es wird über einen Emitterfolger einem IC

	a	c
1	+5V	+5V
2	GND	SCLK
3	"	SDA
4	"	n.c.
5	"	EXD
6	"	/VCR
7	"	n.c
8	"	n.c
9	"	n.c
10	"	n.c
11	"	n.c
12	"	+12V
13	"	n.c
14	"	BAS (VT)
15	"	Switch In Text
16	"	SYNC
17	"	n.c
18	"	VIDEO IN
19	"	Switch In Video
20	"	GND
21	"	GND
22	"	Switch Out
23	"	n.c
24	"	n.c
25	"	n.c
26	"	n.c
27	"	/COR
28	"	B
29	"	G
30	"	R
31	"	Y
32	"	GND

Bild 7. Die Steckerbelegung

TEA5114 zugeführt. Bei diesem Baustein handelt es sich um einen preisgünstigen dreifachen Videoschalter. Beide Eingänge und der Ausgang jedes Schalters stehen an separaten Pins zur Verfügung. Nur einer der Schalter wird genutzt.

Der Controller mit serieller Schnittstelle

Wie schon erwähnt, steuert der Controller die Decoder-Bausteine. Er analysiert die vom Terminal kommenden Befehle und setzt sie in die Sprache des EURO CCT um. Außerdem ermöglicht er den Datentransport zwischen Decoder und Terminal. Der Controller besitzt zwei Schnittstellen: Mit dem Decoder kommuniziert er über den I^2C-Bus, mit dem Terminal ist er über eine V.24-Schnittstelle verbunden. Wir haben eine Low-Cost-Lösung für den Controller getestet, die sowohl die I^2C-Bus-Schnittstelle als auch die V.24-Schnittstelle durch Software realisierte. Diese Lösung hatte den Nachteil, daß die Übertragungsrate der V.24-Schnittstelle nur maximal 300 Baud betragen konnte. Aus diesem Grund haben wir einen Controller aufgebaut, der für die V.24-Schnittstelle eine Hardware-Lösung verwendet.

Bild 4 zeigt die gesamte Schaltung des Controllers. Als Mikroprozessor wird ein Typ 8031 eingesetzt. Die Software ist in einem EPROM 2764 (bzw. 27128) abgelegt. Der restliche Speicherplatz steht für Erweite-

Abkürzungen, Begriffe

ARD
Arbeitsgemeinschaft der öffentlich-rechtlichen Rundfunkanstalten der Bundesrepublik Deutschland, „Dachorganisation" der Landesrundfunkanstalten, der Bundesrundfunkanstalten und als Gast der RIAS Berlin mit beratender Stimme.

BAS-Signal
Fernsehsignal, bestehend aus Leuchtdichte (Luminanz), Austastung und Synchronsignalen (enthält kein Farbsignal).

BBC
British Broadcasting Corporation; bis 1954 war die 1927 gegründete BBC die einzige Rundfunkanstalt Großbritanniens. Die BBC hat eine öffentlich-rechtliche Organisationsstruktur.

BOX-Betrieb
In dieser VIDEOTEXT-Betriebsart werden anstelle einiger Fernsehbildzeilen Texte eingeblendet. Die Textzeilen erscheinen auf dunklem Hintergrund.

CCT
Valvo-Bezeichnung, bedeutet Computer-Controlled Telext chip

CEEFAX
Teletextsystem der BBC; abgeleitet von „to see facts"

FBAS-Signal
Fernsehsignal, bestehend aus Farbart (Chrominanz), Leuchtdichte (Luminanz), Austastung und Synchronsignalen

full chanal mode
In Deutschland bisher nicht genutzte Betriebsart von VIDEOTEXT, bei der die Textinformation in allen Bildzeilen anstelle der Bildinformation übertragen wird. Damit ist es möglich, ein VIDEOTEXT-Programm mit einer sehr großen Seitenzahl zu übertragen, jedoch mit der Einschränkung, daß nun nicht mehr parallel ein Fernsehbild gesendet werden kann. Diese Betriebsart stellt hohe Anforderungen an den Videoprozessor.

Gost rows
Für zukünftige Erweiterungen stehen zusätzliche Zeilen zur Verfügung. In ihnen können sowohl sichtbare als auch unsichtbare Zeichen übertragen werden.

IBA
IBA (Independent Broadcasting Authority) ist die Dachorganisation, unter der die privaten Fernsehgesellschaften Großbritanniens, Independent Television (ITV), seit 1954 zusammengefaßt sind.

IRT
Institut für Rundfunktechnik GmbH, München; Forschungs- und Entwicklungsinstitut von ARD und ZDF.

MIX-Betrieb
Betriebsart von VIDEOTEXT, bei der die Textinformation zusammen mit dem Fernsehbild betrachtet werden kann. Das Fernsehbild dient als Hintergrund für den VIDEOTEXT.

I^2C-Bus
Bedeutet Inter-IC. Bei diesem Bus handelt es sich um einen bidirektionalen, aus nur zwei Verbindungen bestehenden Multimasterbus, auf dem die Daten seriell übertragen werden. Er verbindet die hochintegrierten VIDEOTEXT-Bausteine untereinander.

ORACLE
Optional Reception of Announcements by Coded Line Electronics; von der IBA entwickeltes Teletextinformationssystem.

RGB-Signal
Fernsehsignal, das bereits in die Signale Rot, Grün und Blau zur Bildröhrenansteuerung aufbereitet wurde.

SCL
Bezeichnung der Leitung für die serielle Übertragung des Taktsignals (Clock) des I^2C-Bus.

SDA
Bezeichnung der Leitung für die serielle Übertragung der Daten des I^2C-Bus.

SFB
Sender Freies Berlin, einer der ARD-Anstalten.

Statuszeile
Ist eine VIDEOTEXT-Zeile, die üblicherweise am oberen Bildrand angeordnet ist. Sie enthält Informationen, wie z. B. die Nummer der gerade gezeigten Seite, Uhrzeit und Namen des VIDEOTEXT-Programms.

VIP
Valvo-Bezeichnung, bedeutet Video Input Processor.

VHF-Sender
Very high frequencies = sehr hohe Frequenz, gemeint ist der Frequenzbereich von 30 bis 300 MHz. In diesem Bereich befinden sich die Fernsehkanäle 2 bis 12.

VT
Im Beitrag verwendete Abkürzung für VIDEOTEXT.

WDR
Westdeutscher Rundfunk, die größte ARD-Anstalt mit mehr als 4000 Beschäftigten

ZDF
Zweites Deutsches Fernsehen, strahlt sein Programm bundesweit aus.

STUCKLISTE COMPUTERGESTEUERTER VIDEOTEXTDECODER

HALBLEITER

IC1	SAA 5240	Euro-CCT
IC2	SAA 5230	Videoprozessor
IC3	HM 6264 LP-15	RAM(150 ns.)
IC4	TEA 5114	Videoschalter
IC5	MAX 232	V.24 Treiber
IC6	8031	Mikrocontroller
IC7	74LS373	Latch
IC8	2764/128	EPROM
D1-6	BAW 62 (1N4148)	
T1-10	BC 547	
T11	BF 324	

KONDENSATOREN

		RASTER
C1,9,14	15 pF	5 mm.
C2	1 nF	5/7.5 mm
C3	470 pF	5/7.5 mm
C4,25	22 nF	7.5 mm
C5	270 pF	5/7.5 mm
C6	100 pF	5/7.5 mm
C7	68 nF	7.5 mm
C8	220 pF	7.5 mm
C10	27 pF	5/7.5 mm
C11	15 uF/16 V	
C12	22 pF	5/7.5 mm
C13	22 pF Trimmer	
C15,16	47 nF	7.5 mm
C17	2.2 µF/16 V	
C18	10 nF	5 mm
C19	220 nF	7.5 mm
C20	4.7 µF/6 V	
C21,22,23,33,34	100 nF	4 * 5 mm,1 * 7.5 mm
C24,27,28	47 µF/16 V	
C26	10 µF/6 V	
C29,30	22 µF/16 V	
C31,32	27 pF	5 mm
C35***	220 nF	7.5 mm
C36***	3.3 µF/16 V	

QUARZE

Q1	13.8750 Mhz
Q2	6.0000 Mhz
Q3	11.0592 Mhz

ARRAYS

A1,A2	5 * 1.5 k	6 pol.
A3	5 * 100 Ohm	10 pol.
A4	5 * 470 Ohm	6 pol.

WIDERSTAENDE

R1	56	k
R2	1.2	k
R3	330	Ohm
R4,22,28.29	4.7	k
R5,6,7,8	39	Ohm
R9	15	Ohm
R10,11	2.2	k
R12	220	Ohm
R13,14,17,20,25	1	k
R15,26	68	Ohm
R16	270	k
R18	820	Ohm
R19	3.9	k
R21,27	8.2	k
R23	1.8	k
R24	47	k
R30***	3.3	k
R31***	330	Ohm

SONSTIGES

L1	15 uH
2	IC-Sockel 16 pol.
1	IC-Sockel 20 pol.
3	IC-Sockel 28 pol.
2	IC-Sockel 40 pol.
1	5 pol. Stiftleiste einreihig
1	15 pol. Stiftleiste einreihig
1	64 pol. Stecker

*** Hinweis: Diese Bauteile werden nur für eine Spezialversion der Platine benötigt Außerdem ist Q3,R3 durch eine Spule 50 uH zu ersetzen und Pin 10,IC2 auf GND zu legen.

Bild 8. Bauteilliste

rungen zur Verfügung). Über Port-1, Bit 6 und 7 ist der Mikroprozessor mit dem Decoder verbunden. Bit 7 ist die Daten- und Bit 6 die Taktleitung für den I^2C-Bus. Bit 0 bis 5 von Port 2 dienen als Adressen A8 bis A13 für das EPROM. Die Adressen A0 bis A7 gelangen über ein Adress-Latch LS373 vom Port-Bus des Mikroprozessors zum EPROM. An den gleichen Pins des Port-Bus sind die Datenleitungen des EPROMs angelegt. Über Port 3, Bit 0 und 1 gelangen die Daten vom Mikroprozessor zum V.24-Treiber MAX 232. An dessen Ausgängen stehen dann V.24-gerechte Signalpegel zur Verfügung. Unserer Schaltung stellt die Übertragungsrate in Abhängigkeit vom angeschlossenen Terminal (oder Computer) im Bereich von 300 bis 9600 Baud selbsttätig ein. Pin 1 bis 15 des Mikroprozessors liegen zusätzlich für eventuelle spätere Erweiterungen auf Pfostenstecker auf. Nun noch die Bedeutung einiger Prozessorsignale:

ALE =>
address latch enable; IC LS373 wird angesprochen

EA =>
external access; externe Programmsteuerung, liegt dieser Pin auf +5V, erfolgt die Prozessorsteuerung durch ein Programm im externen Speicher

PSEN =>
program store enable; EPROM wird angesprochen

Besondere Vorteile des Mikroprozessors 8031:

- preiswert
- eingebaute V.24-Schnittstelle
- umfangreicher Befehlssatz, dadurch komfortabel zu programmieren
- großer Adressbereich
- kompatibel zu einem Chip, der auch eine I^2C-Bus-Schnittstelle eingebaut hat, aber wegen des höheren Preises von uns nicht verwendet wurde.

Der Aufbau

Abschließend noch einige Bemerkungen zum Aufbau der Hardware: Alle Bauteile lassen sich auf einer einzigen Europa-Karte (doppelt kaschiert) unterbringen (siehe *Bild 5*, „Layout" und *Bild 6*, „Bestückungsseite"). Alle für die Funktion der Schaltung benötigten Leitungen liegen auf einer 64poligen Steckleiste an. Wie bereits beschrieben, beinhaltet der Baustein TEA 5114 drei Videoschalter mit jeweils zwei Eingänge und einem Ausgang. Alle Ein- und Ausgänge liegen frei zugänglich auf der Steckerleiste. Sollen diese genutzt werden, müssen sie mit kurzen Brücken auf der Steckerleiste mit der übrigen Schaltung verbunden werden. Die genaue Belegung entnehmen Sie bitte *Bild 7*. Zusätzlich sind für eventuelle zukünftige Erweiterungen einige Leitungen auf Pfostenstecker geführt. Die Bauteileliste (*Bild 8*) dient als Referenz zwischen Bestückungsplan und Schaltbildern. Bausätze, fertigbestückte Platinen und fertige Geräte werden in nächster Zeit von zwei Firmen angeboten werden. Bitte beachten Sie gegebenenfalls die Angebotsseiten in mc.

Die Software

Wie schon erwähnt, hat die Software einen Umfang von ca. 2 KByte. Sie ist modular aufgebaut. Das Hauptprogramm bearbeitet folgende Funktionen:

Bild 9. Initialisierungsmeldung

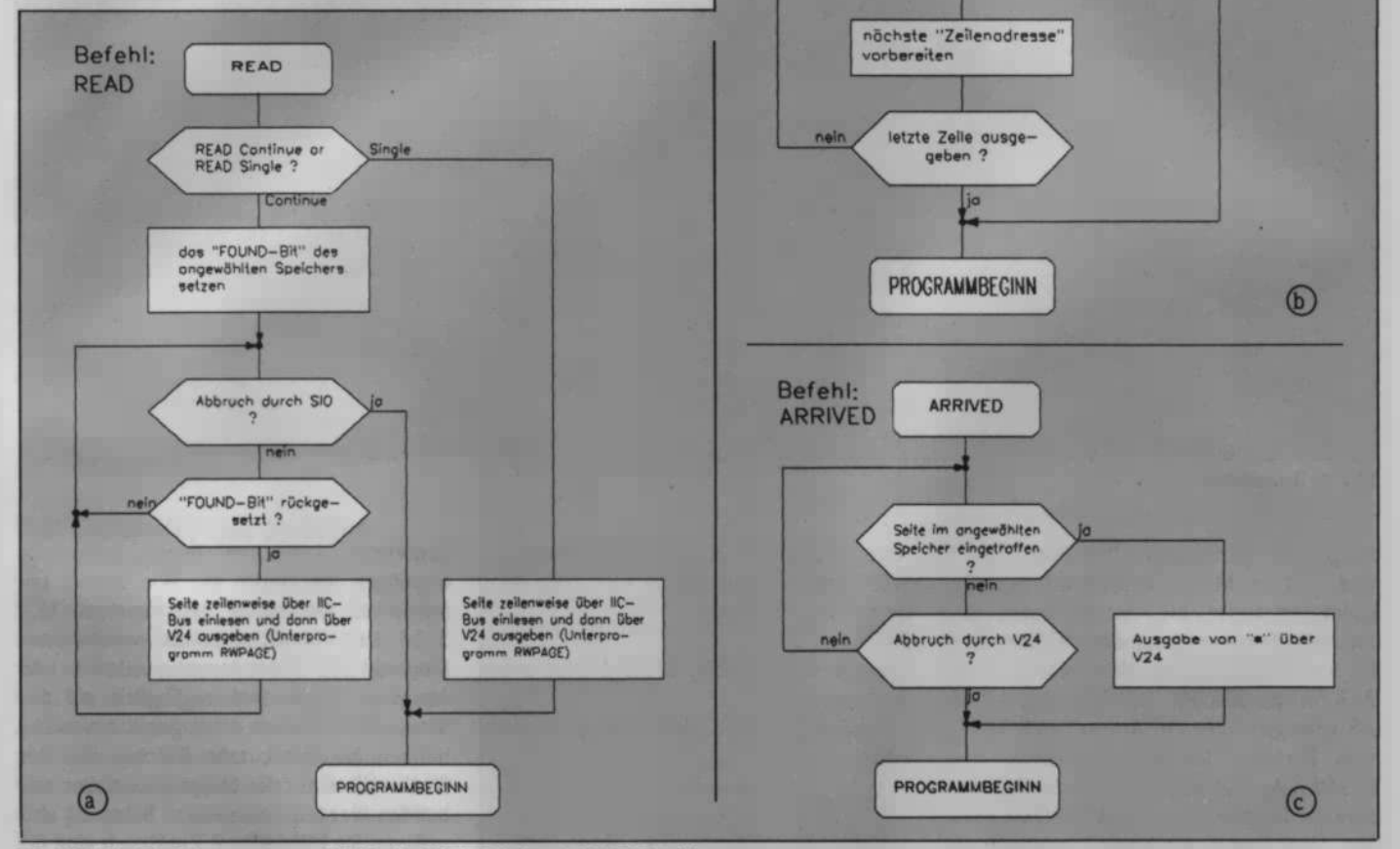


Bild 10. Die Flußdiagramme zu READ (a), WRITE (b) und ARRIVED (c)

Initialisierung:
Der CCT wird initialisiert.
Der deutsche Zeichensatz wird gewählt (über den WI-Befehl können Sie beim Baustein SAA 5240 A auf den englischen oder skandinavischen Zeichensatz umschalten)
Die Flagregister werden gelöscht.
Beim V.24-Baustein wird ein Hardware-Reset ausgelöst.

Programmbeginn:
Die Begrüßungsmeldung (siehe *Bild 9*) wird an den Fernsehbildschirm gesandt (Hinweis: laufen in der Kopfzeile die Nummern der VT-Seite durch, wird ein VT-Signal empfangen!)
Die über das angeschlossene Terminal eingegebene Anweisung wird über die V.24-Schnittstelle eingelesen.
Die Anweisung wird im RAM abgespeichert.

Check:
Umwandlung der eingelesenen Zeichen von Klein- in Großbuchstaben.
Vergleich der Daten mit den Elementen der Anweisungstabelle.
Ist das Zeichen enthalten, den zugehörenden Programmteil ausführen. Kann das Zeichen nicht in der Tabelle gefunden werden, die Fehlermeldung „#" über die V.24-Schnittstelle ausgegeben.

Das kommentierte komplette Quellprogramm kann wegen seiner Länge von mehr als 30 Seiten nicht abgedruckt werden.

Bei der Redaktion kann jedoch die Anschrift des Softwareentwicklers angefragt werden, von dem der komplette Assembler-Quellcode gegen einen Unkostenbeitrag von 20 DM bezogen werden kann.

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

LP 2002

Fotoplot – Zusatz

Mit dem FOTOPLOT-ZUSATZ 'LP 2002' kann jeder handelsübliche Plotter auch als FOTOPLOTTER verwendet werden. Damit wird es z.B. möglich, Platinenlayouts im Maßstab 1 : 1 direkt auf Film zu plotten. Eine Programmsteuerung sorgt für optimale, geschwindigkeitsangepasste Belichtung. Bei einem Gesamtpreis von 1950,–DM (zzgl. Mwst.) ist das Gerät auch für Kleinanwender interessant. Die Erprobung ist gegen einen verrechenbaren Unkostenbeitrag von 150,–DM möglich.
Fordern Sie Unterlagen und Arbeitsproben an ! Distributoranfragen erwünscht !

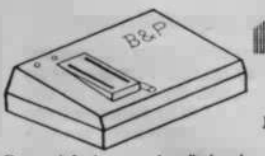
ing.- büro oberbeck
4920 Lemgo
achternstr.117
05261/15480

MINIPREISE FÜR LAUFWERKE

PHILIPS X3132	2 x 40 Spur slim line	DM	209,–
PHILIPS X3134	2 x 80 Spur slim line	DM	240,–
	mit Umschaltung 40/80 Spur	DM	255,–
PHILIPS X3113	1 x 80 Spur 2/3 Bauhöhe	DM	95,–
	mit Umschaltung 40/80 Spur	DM	110,–
PHILIPS X3114	2 x 80 Spur 2/3 Bauhöhe	DM	209,–
	mit Umschaltung 40/80 Spur	DM	229,–
Floppygehäuse für slim line		DM	25,–
Testaturen ohne Gehäuse		DM	25,50
Datenkabel für 2 Laufwerke		DM	32,–
Anschlußstecker für Stromversorgung		DM	2,95

Alle Preise zuzg. Versandkosten. Versand per NN oder Vorkasse

CHRISTEL VON DER LINDEN 4200 OBERHAUSEN
STERKRADER STR. 189 TEL. 0208/663721 AB 14 UHR



mc

EPROM - Brenner

Neue Software
Jetzt 27512 / 27513

Das einfach zu handhabende Programmiergerät zum günstigen Preis. Anschluß an die Centronics-Buchse, daher keine Steckkarte im Rechner erforderlich. Stromversorgung über eingebautes Netzteil. Vollständig über Software gesteuert. Detaillierte Beschreibung in der mc 1/87. Platine doppelseitig durchkontaktiert.

Neue Software-Version 2.10 : Software für PC-/XT/AT/Schneider-PC in Turbo Pascal 3.0 incl. Quellcode. Enthält die EPROM-Typen 2716-27512/27513. Programmiert 2764 in typ. 45 Sekunden (AT). Standardalgorithmen implementiert. Zugriff auf DOS-Ebene aus laufendem Programm. Erstellen und Ausführen von Makros.

Platine, Software	DM 89.-
Bausatz (ohne Gehäuse), Software	DM 249.-
Fertiggerät, Software	DM 549.-
Software	DM 25.-
Gehäuse (unbearbeitet)	DM 29.-

Versandkosten: Ausland DM 12.- Inland DM 8.-
Lieferung mit Handbuch. Versand per Nachnahme.

B & P
B. Seng Ludwig-Dürr-Straße 10 7320 Göppingen
Telefon: 07161/75245

INTERFACES

XON/XOFF↔RTS/CTS

- Datenübertragungstester
- Galvanische Trenn-Adapter (1000 V)
- High-Speed-Adapter (10 MB/s)
- Störsichere Übertrager
- Long-distance-Interfaces (20 km)
- Datenformat-Wandler (Asynchron)
- Baudrate Konverter
- seriell ⟷ parallel-Wandler
- Kurzstreckenmodems > 100 Kbaud
- Netzwerk-Adapter
- Microprozessorgesteuerte Interfaces
- Mehrweg Druckpuffer 256 KB
- Kundenspezifische Interfaces

RS232C (V24, V28)	RS423 (V 10)
Current-Loop (20, 40, 60 mA)	TTL
RS422 (V11) RS485	CENTRONICS
PIN 1360 (X.27, TR30.1)	IEEE 488 / IEC / HP-IB
	BCD

Dipl. Ing. Jürgen Knauft
SOFTWARE HARDWARE DIGITALELEKTRONIC
D 6457 Maintal 1 Birkenweg 1 06181/45643
SOFT

HARDWARE-MESSWERTERFASSUNG

– IBM XT/AT – CBM – hier einige Auszüge IBM –

	ab DM
IEEE-488-(IEC-Bus-)Platine und Software	480
32-Bit-Optokoppler-Input-Platine	480
32-Bit-Optokoppler-Output-Platine	480
8-Bit-32-Kanal-A/D-Wandler, 8 µs Wandlung	560
8-Bit-32-Kanal-A/D-Wandler mit RS-232-Schnittstelle	860
12-Bit-32-Kanal-A/D-Wandler, 25 µs Uni-/Bipolar	560
12-Bit-4-Kanal-D/A-Wandler, 8 µs, Settingtime	350
12-Bit-Input-/Output-Platine, 8/4 Differenz programmierbar, gesockelt	540
12-Bit-I/O-Platine mit 12 Relais, 1x um, 220 VAC, 3 A/80 W + 12x TTL-I/O	560
8fach-RS-232-Umschaltplatine (Umschaltung programmierbar od. Alt+1...8)	760
Multiboard 86, 12 Bit-32 K A/D + 12 Bit-2 K D/A + 96 TTL-I/O, gesockelt	1675
Multiboard 87, 8 Bit-32 K A/D + 8 Bit-4 K D/A + 80 TTL-I/O	980
Thermoboard 86, –50 °C–+150 °C, 16 Kanäle, od. d. –50 °C–+1150 °C	980
Centronics-IEC-Interface (f. Drucker, Plotter mit IEC/IEEE-488)	295
Programmierbarer Timer-Counter, 8fach/24fach, 16 Bit	300/DM 680

NEU IM PROGRAMM

Elektrostatische Plotterauflagefolie zum Festhalten des Plotterpapiers	DM 295
16-Bit-A/D-Turboboard, 16 µs, mit 128 KByte RAM	DM 1850
16-Bit-A/D-Turboboard, 50 kHz, 128 KByte RAM	DM 1290
386-Board, 20 MHz, 1 MB RAM	DM 980 – DM 1453 – DM 2137
32-Bit-Counter-/Kollektor-Output-Platine, max. 50 V/500 mA	DM 390
PT-100-Verstärker für versch. Bereiche, 100 °C, 200 °C...800 °C	DM 150
16-Bit-8-Kanal-A/D-Wandler (Dual-Slope)	DM 920
8048-Emulator-Kit inkl. 2-Pass-Assembler	DM 1190
Einstellbare Quarzzeitbasis	DM 280

Info kostenlos! **LOTHAR BOCKSTALLER**
Hard- und Software – Hadwigstr. 16, 7867 Wehr 2, Telefon 0 77 61/18 08

Preis-Sensation

DIN-A3-Plotter

mit Papierbewegung

TSS 860

HP-GL-kompatibel

DM 3200.–

6 Farben
0,025 mm Auflösung
400 mm/s Zeichengeschwindigkeit
Centronics- und V.24-Interface
56 Zeichenbefehle

Lieferung per Nachnahme

TSS-Schmitz
In der Holl
5223 Bierenbachtal · Tel. 0 22 93/21 88
Inh. Brigitta Schmitz

mc 4/88

Cross-Assembler

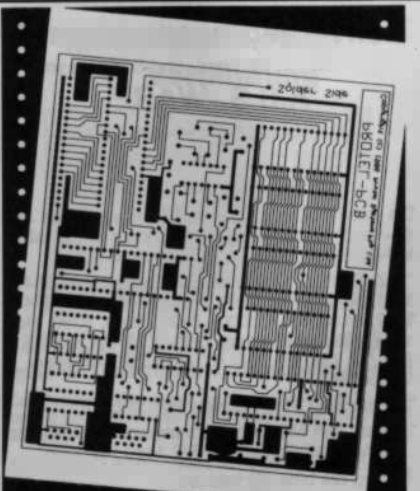
Professionelle Cross-Assembler für PC-DOS/MS-DOS und VAX-Systeme, MACROS, Linker, Library-Manager, Conditionals, unbegrenzte Programmlänge, verschachtelte Strukturen, schnelle Compilation, zahlreiche Zielformate.

Unterstützte Prozessor-Familien:

Z8, Super-8, Z80, Z180, 64180, Z280, Z8000, Z80.000, 8080, 8085, NSC800, 8088/86, 80186/286, 8048, 8044/51, 80452, 80515, 8096, 83c351, 8400, 1802, 6301, 6800/02/08, 6801/03, 6804, 6805, 6809, 68000/8/10, 68020, 65c01/02/11, 65c816, 32000, V20/V30, F8/3870, NEC7500, NCR32

Weiterhin liefern wir alle Tools für den Entwickler: C-Compiler, Simulatoren, In-Circuit-Emulatoren, Programmer, Prüf- u. Meßgeräte.

IDA Systeme!
Ingenieurbüro für Datenverarbeitung
Dipl.-Ing. Hans-Jürgen Wilke
Adalbertsteinweg 26 D-5100 Aachen
Tel.: 0241/542228 FAX: 0241/533230



IDA Systeme!
Ingenieurbüro für Datenverarbeitung
Dipl.-Ing. Hans-Jürgen Wilke
Adalbertsteinweg 26 D-5100 Aachen
Tel.: 0241/542228 FAX: 0241/533230

Leiterplatten-LAYOUT

Professionelles Leiterplatten-Entwicklungssystem für PCs:

- **Schaltplan-Erstellung**
- **Interaktives Routen**
- **Auto-Routing**
- **1/1000 Zoll Auflösung**
- **1–8-Lagen-Multilayer**
- **480x810-mm-Boards**
- **SMD-fähig**
- **Komfortable und schnelle Arbeitsweise**
- **Outputs f. Photo- u. Stiftplotter, Matrix-Drucker**
- **Laufende Programmpflege/ Update-Service**

Demo-Version (CGA/EGA, Mouse) 30.– DM

IDA Systeme!
Ingenieurbüro für Datenverarbeitung
Dipl.-Ing. Hans-Jürgen Wilke
Adalbertsteinweg 26 D-5100 Aachen
Tel.: 0241/542228 FAX: 0241/533230

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

GWK GESELLSCHAFT FÜR TECHNISCHE ELEKTRONIK mbH

NEU
68020 – 68881
für
Selbstbaucomputer
c't 68000 GWK

Weltweit leistungsfähigstes Systemkonzept für modulare Systeme im Einfach-Europa-Format

Kostenloses Informationsmaterial anfordern!

Postfach 1360, D-5120 Herzogenrath,
Telefon (0 24 06) 60 35, Telex 8 32 109 gwk d

DER NEUE INDUSTRIE-COMPUTER IM DOS-STANDARD

- als intelligente Steuerung von Produktionsprozessen, speziell mit Kopplung an speicherprogrammierbare Steuerungen
- zur Qualitätskontrolle, Datenerfassung und Dokumentation
- ihren Anforderungen entsprechend auslegbar bzw. ausbaufähig durch Standard-Bus-Slots (kein Spezial-Bus!)
- mit AT-/XT-CPU, div. I/O-Karten: digital 24 V, analog, seriell
- mit neuer RAM-/EPROM-Karte, z. B. zum Betrieb ohne Laufwerk, einfache Programmübertragung in die Eproms
- mit XT-SUB-CPU mit digitalen, analogen und seriellen I/O, DOS 3.2 und Pascal im Eprom
- mit neuer, gegen Staub und Wasser geschützter Industrietastatur
- im kompakten, robusten Gehäuse für den Schaltschrankeinbau
- alles ist kompatibel zum Industrie-Standard
- mit Treiber- und Betriebssoftware
- mit der kompletten Software für Ihr gesamtes Projekt
- mit Beratung, Projektierung und Inbetriebnahme

Rufen Sie uns an oder schreiben Sie uns.
Wir informieren Sie gerne ausführlicher.

GRIMBERG INDUSTRIE-ELEKTRONIK
Kölner Str. 246–248, D-5275 Bergneustadt,
Tel. 02261/4691, Telex 884854 gse d

SOFTWARE & HARDWARE
für Apple II & Mac & MS-DOS-Computer

z. B. für APPLE II und Kompatible
Wir liefern die RAMworks-Karte (AE) für den Apple IIe mit max. 16 MB (z. B. erweitert Appleworks) z. B. RAMWORKS III 64 K ..... DM 469.–
Anpassung für Appleworks 1.2 auf dem II+/e.
Original oder mit externer Tastatur. Anpassung in deutsch
für SATURN 128 K und IBS AP33 1 MB! ..... DM 130.–
AD 16 Ch. 12 Bit, schnell! (Applied Eng.) ..... DM 799.–
PKASO/U-Printer-Karte (IS) ..... DM 429.–
Timemaster II H. O., die Uhrenkarte! (AE) ..... DM 299.–
ELF kompl. Statistik-Software (TWG) ..... DM 495.–
Prime-Plotter-Grafik-Software (Primesoft) ..... DM 629.–
Z-RAM Ultra 3, 256 K, Z80 CP/M + Uhr, max. 1 MB ..... DM 798.–
Z-RAM Ultra 2, 256 K, mit Uhr, max. 1 MB ..... DM 698.–

z. B. für IBM und Kompatible
MatchPoint-PC plus UNIFORM-PC (Lesen/Schreiben von Apple Disks im IBM PC/XT/AT/100 % komp. für DOS-ProDOS-CP/M) ..... DM 649.–
MatchMaker (Anschluß von Mac 800 K Drive an IBM PC/XT/AT/100 % komp. zum Datentransfer Mac-MSDOS) ..... DM 498.–
CompatiCard (Anschluß von 3.5", 5.25", 8" Drives an IBM PC/XT/AT/100 % komp. z. B. 3.5" 720 K/1.44 MB) ..... DM 629.–
XENO-PC (Cisin) (Lesen/Schreiben von CP/M- u. MS-DOS-Formaten im IBM PC/XT/AT/100 % komp.) ..... DM 258.–
ELF PC kompl. Statistik Software (TWG) ..... DM 500.–

z. B. für alle Systeme
Printerchanger 3 parall. Drucker auf 1 Micro
inkl. Kabel/Netzteil (Keyzone) ..... DM 340.–
Printersharer 3 Micros auf 1 parall. Drucker
inkl. Kabel/Netzteil (Keyzone) ..... DM 340.–

Wir sind Import-Spezialisten und bieten Ihnen eine große Auswahl an Software und Hardware bedeutender Hersteller aus den USA und England.
Informationen gegen DM 3.– in Briefmarken.

WEISS COMPUTER Dipl.-Psych. Karl-Heinz Weiß
Am Wiesenhof 17, 2940 Wilhelmshaven, Tel. 0 44 21/8 31 79

Buchhaltung mit BUCH

Doppelte Buchführung, die auf Selbständige und Kleinunternehmer zugeschnitten ist:

- **14 Tage frei zur Ansicht**
- einfache und sichere Benutzerführung
- Prüfsiegel „anwenderfreundliche Software"
- Betriebsübersicht, G&V, Debitoren, Creditoren, Journal, Kontenblätter, Saldenliste, Bilanz usw.
- max. 32 000 Buchungen, 1200 Konten
- automatische Umsatzsteuerberechnung
- **Interface für DATEV, DBASE usw.**

... und so urteilt die Presse:

„einfach zu bedienendes Programm, das dem Einsteiger entgegenkommt" (CHIP 6/87)
„... ein leicht zu beherrschendes Finanzbuchhaltungsprogramm für kleinere Betriebe, das sehr anwenderfreundlich konzipiert wurde" (c't 7/87)
„... ein leistungsfähiges, leicht zu bedienendes Werkzeug" (PC + Soft 7/85)
„halten wir dieses Finanzbuchhaltungsprogramm für ganz ausgezeichnet" (Softwaretestjahrbuch 86)

180.– bis 680.– DM für PC/XT/AT, Macintosh, II+,c,e,gs

RÖNTGEN SOFTWARE
Oltmannsstr. 34, 7800 Freiburg, Tel. 07 61/40 87 40

Das kostenlose Christiani Kursprogramm

Lernen Sie heute, was morgen unentbehrlich ist.

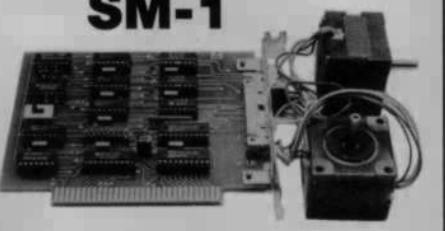
Fordern Sie gleich das kostenlose Kursprogramm an, das Sie über unsere Lehrgänge informiert – u. a. auch über:

- ☐ Elektronik-Labor
- ☐ IC-Labor
- ☐ Digital-Labor
- ☐ Oszilloskop-Labor
- ☐ Fernseh-Labor
- ☐ Elektronisches Messen
- ☐ Mikroprozessortechnik
- ☐ Peripherie-Bausteine
- ☐ Kompakt-Kurs BASIC
- ☐ BASIC + Mikrocomputerpraxis
- ☐ Amateurfunk
- ☐ Automatisierung
- ☐ Optoelektronik-Labor
- ☐ El. Steuerungstechnik
- ☐ Videotechnik
- ☐ Kompakt-Kurs Elektronik
- ☐ CNC-Technik
- ☐ C64 CNC-Simulator
- ☐ Bautechnik
- ☐ EDV-Grundlagen

Gewünschtes ankreuzen; Anzeige ausschneiden und einsenden an

Dr.-Ing. P. Christiani · Technisches Lehrinstitut
Postf. 35 69160 · Tel. 0 75 31/58 01-0 · 7750 Konstanz

SM-1



Die Schrittmotorenkarte für XT, AT

- 2 bipolare Schrittmotoren anschließbar
- 8-Bit-Eingabeport (TTL-Pegel)
- 6-Bit-Ausgabeport (o. K. 500 mA/50 V)
- Motorspannung intern/extern umschaltbar
- Stromreduzierung für Ruhezustand

Steckkarte aufgebaut u. getestet
mit 2 Motoren 3,6°/1,8° 12 V DM 189,–
Versand per Nachnahme

Soft- und Hardware Tetzlaff
Grotenbachstraße 9, 4600 Dortmund 50
Telefon 02 31/71 30 42 ab 15.30 Uhr

SOFTWARE THAT SPEAKS YOUR LANGUAGE

Multi-Lingual Scholar 3.0

Ein Textverarbeitungsprogramm, das gleichzeitig mehrere Sprachen und verschiedene Alphabete in einem Text zuläßt und diesen in mehreren Schriftarten und -größen in ausgezeichneter Qualität ausdruckt.
Die Laufrichtung wird selbstverständlich berücksichtigt: Arabisch/Farsi, Hebräisch rechts nach links, Lateinisch, Griechisch, Cyrillisch links nach rechts.
Darüber hinaus können Sie mit dem zugehörigen FontGenerator auch selbst beliebige Schriften/Alphabete erzeugen.

- Standard-Ausgabe: 5 Alphabete, je 2 Schriften ROMAN (alle latein. Schriften), GREEK (griech./math.), CYRILLIC (russ., bulgar. usw.), HEBREW (hebr., aram.), ARABIC (arab., pers. usw.)
  Treiber für 9- u. 24-Nadel-Drucker, engl. Dokumentation
  Preis exkl. MwSt.: öS 5300,–
- Laser-Option: wie Standard, zusätzlich Treiber für Laser-Drucker (HP Laser Jet, Corona) Preis exkl. MwSt.: öS 6990,–
- Demo-Version: ähnl. Standard, jedoch limitierte Kapazität (2000 Zeichen, 2-Zeilen-Druck) Preis exkl. MwSt.: öS 300,– (wird angerechnet)

Versand per NN, Kosten öS 60,– (Österreich), öS 100,– (Ausland). Händleranfragen gerne erwünscht.

MEC GmbH
Favoritenstr. 22, A-1040 WIEN, Tel. (2 22) 65 38 56

Konvertierungsprobleme?

lösen Sie mit der Hard- und Software von

A.S.S.-WARE

Mainframe *1) <–> MS-DOS
CP/M *2) <–> MS-DOS
MS-DOS *3) <–> MS-DOS

Die Konvertierungssoft- und Hardware läuft auf XT- und AT-kompatiblen Rechnern unter allen MS-DOS-Versionen ab DOS 2.X und 3.X.

*1) Z. B. Basisdatenaustauschformat: IBM 3540, 3601, 3602, 3740, 3741, (= IBM/370) 3742, 3747, 3962, 5110, 5114, 5231, 5320, 5340, 5380, Honeywell Bull, SEL usw.
IBM /32, /34, /36, /38. 8" und 5.25"-Disketten.
Nixdorf 8870

*2) Fast alle CP/M-Formate von 3" bis 8"

*3) Fast alle MS-DOS-Formate von 160 KB bis 2 MB natürlich auch PS/2 1440 KB netto

Fordern Sie bitte ein Info an.

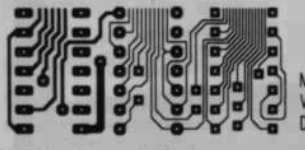
A.S.S.-WARE, Alfred Herrmann, Schimmelshahn,
5461 Roßbach/Wied, Tel. (0 26 38) 45 13

ATARI ST
NEC P5/P6/P7 EPSON LQ800 etc.

PLATINEN-LAYOUT

MPK Printed Circuit Board Editor, Version 3.4 new!

Professionelles, voll mausgesteuertes Leiterplatten CAD Programm ohne Autorouter. Hardwarevorauss.: ATARI ST (MEGA ST) mit 1 Mb und ROM TOS, sowie ein 24nadel Drucker.
Platinen bis 203•240mm, beidseitig. 1/180 Inch Auflösung, Semifeinstleiter, 45 Grad Leiterbahnwinkel.
Gesamtbildfadenkreuzcursor mit einst. Schrittweite, WYSIWYG, real-time-zoom und ein sehr schneller Bildaufbau erleichtern die interaktive Layouterstellung (Progr. läuft nur monochrom).
Jederzeit Top View oder Bottom View (Spiegeln).
Ausdruck mit hoher Auflösung 1:1 für Prototypen oder 2:1 in Produktionsqualität (! garantiert praxis-erprobt !).
Ausfüllen der Lötaugenmittelpunkte beim Drucken möglich.
Getrennter Bohrplanausdruck (Lötstopmaske).
Der MPK PCB Editor kommt vorwiegend im professionellen Bereich zum Einsatz.
MPK PCB EDITOR mit deutsch. Handbuch DM 179,– zuzgl. Vers. per Nachnahme.
Originalausdruck P6:



MPK
Marek Petrik
Vogelsbergstr. 13
D-3550 Marburg 7
06421 / 47588

Händlerkonditionen auf Anfr.!

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

SHARP **PC 1600** SHARP

Wir haben d i e Lösung für Ihre Speicherplatzprobleme.

P C - 1 6 0 0

Unsere Speichererweiterungsmodule werden einfach in Slot 2 bzw. Slot 1 (32 K-Module) geschoben.

RAM-MODULE	normal	extra**
32 K-Byte	159,00 DM	209,00 DM
128 K-Byte	489,00 DM	529,00 DM
256 K-Byte	639,00 DM	689,00 DM

extra** = wahlweise akku- oder batteriegepuffert mit Schreibschutzschalter

E-PROM
32 K-Byte 110,00 DM / Programmieradapter 120,00 DM

P C - 1 5 0 0

Interner Speicherausbau: Grundpreis 136,80 DM plus 22,50 DM pro 8 K-Byte. Bis 568 K-Byte.

Porto/Verpackung zu Selbstkosten. Wir gewähren ein Jahr Garantie.

Achtung !! Neue Anschrift !! Achtung !!
E L K E B A J I C
Planungsbüro
Hardware * Software * Elektrotechnik
Bauer Landstraße 99 2390 Flensburg
Telefon: (0461)420-37/38 Telefax: (0461)45026

SHARP SHARP

Ecosoft Economy Software AG

Postfach 1905, Abt. A13, 7890 Waldshut, Tel. 077 51 - 79 20

Prüf-Software und Frei-Programme (fast) gratis

- **Größtes Angebot Europas: Über 3'000 Disketten für IBM-PC/Kompatible, Macintosh, Amiga, Atari ST, C64/128, Apple II.**
- **Viele deutsche Programme, speziell für Firmen, Selbständige, Privatpersonen, Schulen.**
- **Software gratis, Sie bezahlen nur eine Vermittlungsgebühr von DM 14,40 oder weniger je Diskette.**
- **HOTLINE: Tel. Anwenderunterstützung (3 Techniker von 8-18 Uhr).**

Verzeichnis gratis

Bitte Computermodell angeben.

Komplettlösungen State of the Art

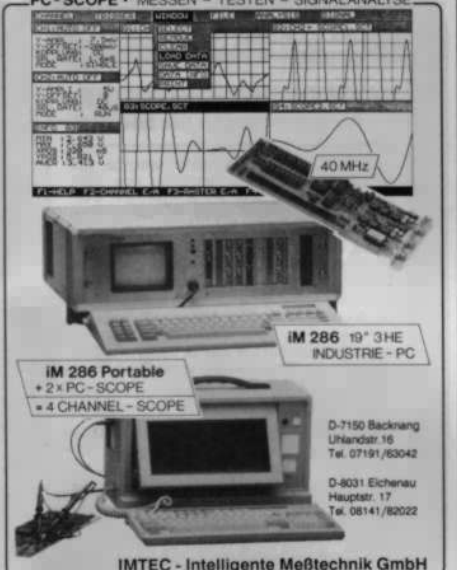
MC modular – AT 0 Waitstates, 8/10/12 MHz, 40 MB, 28-ms-Festpl., 3 MB RAM bestückt, 100 ns, 1 × 1,2-MB-Disk, multitaskingfähig, menügesteuert, DOS 3.2, Softwhite **5995.–**
Aufpreis Tower-Gehäuse **495.–**
Aufpreis Farbe: Super-Level-5-EGA-Karte, EIZO-Flexscan **2595.–**
Aufpreis Ist Publisher, Logimouse **495.–**
Aufpreis 102 MB, 3 ms (Sie haben richtig gelesen, so schnell) ... **3595.–**
Aufpreis 80386er-CPU ..... auf Anfrage
Aufpreis 300-dpi-Scanner + Softw. **2295.–**
Aufpreis Diagraph 2000 ..... **995.–**

XT-Aufrüstungen, Wavemate Bullet

Machen Sie aus Ihrem alten Rechner einen Renner! 80286-XT-Board, 7,2 MHz . **1695.–**
10 MHz, voll bestückt 1 MB **2095.–**
12,5 MHz, voll bestückt 1 MB **2495.–**

lehrer-pool GmbH c/o Capito
Barerstraße 53, 8000 München 40,
Telefon 0 89/2 72 23 99

PC-SCOPE · MESSEN – TESTEN – SIGNALANALYSE



iM 286 19" 3 HE INDUSTRIE - PC

iM 286 Portable
+ 2 × PC - SCOPE
+ 4 CHANNEL - SCOPE

D-7150 Backnang
Uhlandstr.16
Tel. 07191/63042

D-8031 Eichenau
Hauptstr. 17
Tel. 08141/82022

IMTEC - Intelligente Meßtechnik GmbH

Interface zum Digitalisieren von Videobildern (TV, Kamera und Recorder) in 1/50 Sekunde (bei 2 Graustufen) fuer IBM XT/AT.

VIDEO-1000 I 495,-DM
Aufloesung 640 x 200 und 640 x 288 Pixel, max. 32 Graustufen, Unterstuetzt:
CGA 640 x 200 und 320 x 200
HERCULES 640 x 288 Pixel
EGA 640 x 200 und 640 x 288

VIDEO-1000 II 795,-DM
Aufloesung 1280 x 612 und 768 x 852 Pixel, max. 32 Graustufen, Halbbilderkennung, Unterstuetzt:
CGA 640 x 200 und 320 x 200
HERCULES 720 x 348 Pixel
EGA 640 x 200 und 640 x 350

UTILITIES 1 ..... 98,-DM
Utilitie-Diskette mit ca. 30 Programmen. Kurzfilm, 3-D Bilder (Rot/Gruen), Digitalisierung aus Grafikprogrammen, Druckroutine fuer P6 + P7 mit max. 16 Graustufen etc. Bitte bei Bestellung VIDEO-1000 I oder II angeben.

UTILITIES 2 ..... 98,-DM
Utilitie-Disk speziell fuer DESKTOP-PUBLISHING. Man kann Bilder mit 10 verschiedenen Rastern digitalisieren und im PAINTBRUSH-Format bis max. 1536 x 1152 Pixel abspeichern. Diese Diskette gibt es nur fuer VIDEO-1000 II.

Infoblatt gratis. Demodisketten (4 Stck) nur gegen Einsendung von 20,-DM (Schein oder Briefmarken). Der Versand der Digitizer erfolgt p.NN.

Ing.Buero Manfred Fricke Neue Str.13 1000 Berlin 37 Tel: 030/801 56 52

GRAPHIK-ERWEITERUNGEN für:

IBM-PCs
VIDEO-DIGITALISIERER RDDK

Interface zur Bilderfassung und -bearbeitung
Auflösung: 2 Seiten mit 640 × 400 Pix. und 256 Farben, Speicherung 1/25 Sek. pro Vollbild, Zoom, Scroll, Anschluß an Kamera, Videorec. usw., 512 KByte RAM, DMA.
Preis RDDK **4560 DM**

APPLE II
VIDEO-DIGITALISIERER AD16-4

Echtzeitdigitalisierung (1/30 Sek.) bei 256 × 256 Pix. und 16 Graustufen, 128 KByte RAM, BAS und RGB-Ausgang, Anschluß für fremdsynchronisierbare Kameras, 9365-Prozessor, Software auf Disk.
Preis AD16-4 **1710 DM**

DISPLAYKARTE HGR-1024

Prozessor NEC 7220, Bildspeicher für 1024 × 1024 P., 128 KByte, Zoom, Scrolling, Fenster, BAS- oder TTL-Ausgang, erweiterter Befehlssatz für Basic und Demo auf Disk.
Preis HGR 1024-SW **912 DM**
Preis Farbversion (384 K, 8 Farben) **1596 DM**

SONDERENTWICKLUNGEN AUF ANFRAGE

BARTSCHER ELEKTRONIK
Westring 84, 3440 Eschwege, Telefon 0 56 51/17 95

Immer die richtige ⇋ Verbindung

EPSON HX-20 mit ADDA-P2

der wirklich mobile Meßcomputer

Zur einfachen Erfassung und Verarbeitung analoger und digitaler Signale.
(4 bipolare A/D-Kanäle + 8 dig. Ein-/Ausgänge)

ADDA-P2: 790.– DM (900.60 DM inkl. MwSt.)

CSM **Winfried Koch CSM**
Mühlwiesenstr. 1
7024 Filderstadt-Sielmingen
Telefon: 0 71 58/6 48 47

Immer die richtige ⇋ Verbindung

EPSON HX-20 mit Port-BOX

auch m. Timer/Counter im mobilen Einsatz

Das System erlaubt neben dem Anschluß von BCD-Geräten die Ein- und Ausgabe von Impulsen zum Ereigniszählen und Frequenzmessen.

Port-BOX: 890.– DM (1014.60 DM inkl. MwSt.)

CSM **Winfried Koch CSM**
Mühlwiesenstr. 1
7024 Filderstadt-Sielmingen
Telefon: 0 71 58/6 48 47

Immer die richtige ⇋ Verbindung

ADDA-P3 am EPSON HX-20

vereint Port-BOX und ADDA-P2

Damit wird der gleichzeitige Anschluß von analogen Sensoren und BCD-Geräten möglich. Optional kann das ADDA-P3 mit Timer/Counter ausgestattet werden.

ADDA-P3: 1290.– DM (1470.60 DM inkl. MwSt.)

CSM **Winfried Koch CSM**
Mühlwiesenstr. 1
7024 Filderstadt-Sielmingen
Telefon: 0 71 58/6 48 47

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

PLATINEN-LAYOUT

Leiterplatten-Entwurf auf PC/XT/AT-Systemen

BOARDSTAR ist ein Software-Paket für Entwurf und Reinzeichnung gedruckter Schaltungen und bietet Ihnen professionelle Leistungsmerkmale zu einem fairen Preis. Komfortable mausgesteuerte Benutzeroberfläche - Leiterbahnführung im 1/40° Raster - Bearbeitung doppelseitiger Platinen bis 516 qcm - Verwaltung von Macro-Bibliotheken - Reinzeichnung mit jedem handelsüblichen Plotter.

ROUTSTAR ist der Autorouter zum BOARDSTAR und erstellt selbsttätig Layouts aus vorgegebenen Schaltungsbeschreibungen: Layout-Beschreibung durch Grafik- oder Textfiles - Berücksichtigung vorgegebener Designkriterien - Problemlose Verarbeitung von BUS-Strukturen.

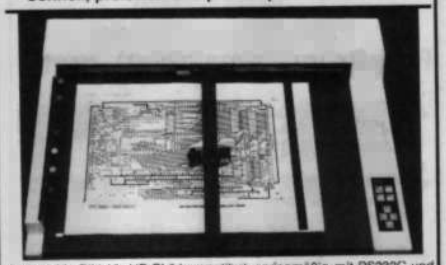
BOARDSTAR (interaktiver Layout-Editor)	698,- DM
ROUTSTAR (Autorouter zum Layout-Editor)	698,- DM
BOARDSTAR-Paket (enthält BOARDSTAR+ROUTSTAR)	998,- DM
Lauffähiges Demo-Paket (wird angerechnet)	70,- DM
DIN-A3 Flachbett-Plotter PL-450 (8-Stift/HPGL/0,025mm)	2450,- DM

Wir bieten Ihnen auch ein breites Spektrum von Service-Leistungen rund um das BOARDSTAR-Programm, vom Plot- u. Repro-Service bis zur Platinenfertigung nach angelieferten BOARDSTAR-Disketten. Fordern Sie Informationsmaterial oder besser gleich das Demopaket an.

Datentechnik Dr.Gert Müller Diezstraße 2A
D-5300 Bonn 1
Tel. 0228/217297

TURBO-PLOTTER

Schnell, preiswert und präzise plotten mit PL-450



Format bis DIN-A3, HP-GL* kompatibel, serienmäßig mit RS232C und Centronics-Schnittstelle ausgerüstet, Pen-Speed maximal 565 mm/s, programmierbarer Stiftwechsel für 8 Farben (Pens), Aufnahme für HP-Standard Faser-/Keramik-/Tusche-Pens, Pen-Magazin mit Automatic-Verschluß, Auflösung 0,025 mm, verschied. Zeichensätze und und.....

Sofort lieferbar... Bitte erfragen Sie den aktuellen Preis!

Datentechnik Dr.Gert Müller Diezstraße 2A
D-5300 Bonn 1
Tel. 0228/217297

* HP-GL ist ein Warenzeichen der Hewlett-Packard GmbH

mtr3, DER EPROM-HELFER

sekundenschnell — automatisch — handlich

- für EPROMs und EEPROMs bis 256Kbit (Option 1Mbit),
- Adapter selbst für wenig gebräuchliche Bauelemente,
- sekundenschnelle aber auch konventionelle Programmierung,
- Typwahl automatisch oder zumindest ohne Datenblatt-Hilfe,
- Emulatorfunktion
- V.24/RS232C-Schnittstelle, einfach konfigurierbar,
- Netz- und Akkubetrieb,

und das alles im kleinsten (189 x 138 x 48) autonomen EPROM-Programmierer für

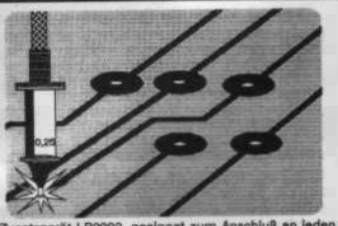
DM 2257,20 **Sfr 1950,— exkl. Wust**

Meßtechnik Dr.-Ing. Ranfft
Dörpfeldstr. 15, D-5657 Haan 2, Tel. 02104/62827

Hedinger + Partner KG
Zentralstr. 3, CH-8610 Uster, Tel. 01/940 90 55

FOTOPLOTTER

Wir machen Ihren Penplotter zum Fotoplotter!



Fotoplot-Zusatzgerät LP2002, geeignet zum Anschluß an jeden Flachbett-Plotter. Gleichbleibende Schärfe und Strichbreite durch geschwindigkeitsabhängige Steuerung der Lichtintensität. Hervorragend geeignet zur Herstellung von Leiterplatten-Filmen mit CAD-Systemen.

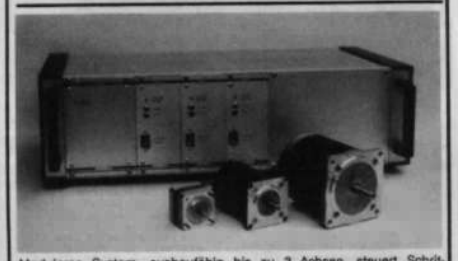
Fotoplot-Zusatz LP2002 .....2223,00 DM

Fordern Sie Informationsmaterial an!

Datentechnik Dr.Gert Müller Diezstraße 2A
D-5300 Bonn 1
Tel. 0228/217297

XYZ-STEUERUNG

Schrittmotoren positionieren unter *HPGL



Modulares System, ausbaufähig bis zu 3 Achsen, steuert Schrittmotoren bis zu 2A Arbeitsstrom - anschlußkompatibel zu ISERT-Anlagen. Der eingebaute Rechner (68000) besitzt eine Centronics-Schnittstelle und ermöglicht so z.B. CNC-Fräsen, -Bohren und Gravieren unter direkter Kontrolle von PC/AT CAD-Systemen wie AutoCAD(tm).

3 Achsen komplett (ohne Schrittmotoren) ..... **1980,00 DM**

Datentechnik Dr.Gert Müller Diezstraße 2A
D-5300 Bonn 1
Tel. 0228/217297

* HP-GL ist ein Warenzeichen der Hewlett-Packard GmbH

PRINT TECHNIK

ÖSTERREICH · 1060 WIEN · STUMPERGASSE 34 · TEL. 0222/597 3423 · TELEX 1 12996

SCHWEIZ · MICROTRON · 2542 PIETERLEIN · BAHNHOFSTR. 2 · TEL. 032/87 2429

VIDEOTEXT-DECODER
für den C64/128 DM 298,–

VIDEOTEXT ist ein Informationsservice, der gleichzeitig mit TV-Programmen von ZDF, ORF, SRG, SKY, RAI etc. ausgestrahlt wird. Mit einem dafür geeigneten Fernseher kann man aktuelle Nachrichten, Sport, Börsenberichte, Flugverbindungen, Wetterberichte etc. abrufen. ABER AUCH DAS VIDEORECORDER-TUNER-SIGNAL kann zum Empfang der Informationen mittels des C64/128er-Computers genutzt werden. Hard- und Software erlauben VIDEOTEXT seitenweise aufzurufen, auszudrucken, auf Diskette zu speichern. Eine sensationelle Neuentwicklung der Print Technik, Wien, für den C64/128-User.

C64/128 VIDEODIGITIZER DM 248,–
Auch für ATARI und IBM erhältlich.

Computerperipherien
8000 München 40 · Nikolaistr. 2 · Tel. 089/36 81 97
Katalog DM 3,– · Tägl. Versand · Telex 5 23 203 d

ELINK-03 Datenfernübertragung ohne Probleme

- 300 BAUD vollduplex
- 1200 BAUD halbduplex
- 1200/75 BAUD vollduplex
- 75/1200 BAUD vollduplex
- ELINK-03 wählt vollautomatisch und nimmt Anrufe entgegen (MAILBOX)
- SMARTMODEM kompatibel: arbeitet mit Standardsoftware wie OPEN-ACCESS, SYMPHONY, KERMIT, CROSSTALK usw.
- Eingebaute Intelligenz erledigt Datentransfer mit jedem Computer oder Terminal über V-24-Schnittstelle, natürlich mit automatischer Baudratenerkennung
- Einfache Programmierung für Spezialaufgaben durch den Anwender
- Betrieb an Nebenstellenanlagen möglich
- Postzulassung mit FTZ-Nummer
- günstige Mengenstaffeln
- komfortables MAILBOX-Programm ebenfalls erhältlich

MICOM Computer- und Informationssysteme Entwicklungs- und Vertriebsges. mbH
Wüstenhofer Str. 6 · 5600 Wuppertal 1 · Tel. 02 02/44 34 01

mc-AllProg Plus

Universelles Programmiergerät aus mc 6-8/1987
mit erweiterter Software

+ Programmiert jetzt auch GALs.
  PROMs von TI und Zeropower-RAMs
+ Vorbereitet für Single-Chip-Prozessoren von INTEL
+ Unterstützt das INTEL-Hex-Format
+ Inhaltsverzeichnisanzeige

- Programmiert (E)EPROMs bis 27512, 27513 und 27011, PROMs von Signetics/Valvo und PALs von MMI, NS, TI und AMD
- Anschluß über RS 232 (V.24) Schnittstelle
- Bediensoftware für IBM-PC/XT/AT (unter PC/MS-DOS), kompatible und Z80-Systeme (unter CP/M) verfügbar
- Bildschirmorientierter (E)PROM-Editor
- Bildschirmorientierter Text-Editor zum PAL-Entwurf
- PAL-Assembler + PAL-Simulator + PAL-Disassembler
- Fuseliste als JEDEC-Dateien speicher-/lesbar

#	Bausatz (inkl. Handbuch und Software)	997,– DM
#	Fertigplatine (getestet: Handb., Software)..	1420,– DM

Versand per Nachnahme. Bei Bestellung unbedingt Diskettengröße, Diskettenformat und Betriebssystem angeben!

Holger Haase & Michael Menrad
Peripheriegeräte GbR
Postfach 2647 / 3300 Braunschweig
Telefon: (0531) 400310

MC BASIC-EMUF

Neu: mc-GAL-Programmer aus mc 2/88, programmiert GAL 16V8 und 20V8 mit BASIC-EMUF oder IBM-PC

Bausatz GAL-Programmer o. TEXTOOL	DM 98,–
Fertigplatine GAL-Programmer o. TEXTOOL	DM 148,–
3M-TEXTOOL-Sockel-Modul, 24polig	DM 49,–
Fertiggerät im Gehäuse für IBM-PC	DM 348,–
Basic-EMUF	
BS1: Platine, GAL, deutsches Manual	DM 98,–
BS2: BS1 + IP80C52, XTAL1	DM 198,–
BS3: RTC58321, AKKU	DM 29,–
BS4: µPD7002, ZN458, XTAL2	DM 29,–
BS5: ULN2003A = 75468, 2× Relais	DM 29,–
BS7: BS1 + TL7705, MAX232, IP80C52, 74HCT08, 74HCT373, 71055, RAM8K	DM 238,–
BSLCDKEY: Bausatz komplett	DM 198,–
FPLCDKEY: Fertigplatine komplett	DM 248,–
Basic-EMUF-Fertigplatine, komplett mit 32 kB RAM	DM 638,–
Basic-EMUF-Fertigplatine, Lochrasterversion mit 8 kB RAM	DM 438,–
PASCAL-CROSS-COMPILER (MS-DOS)	DM 969,–
CROSS-ASSEMBLER (MS-DOS)	DM 869,–
CROSS-SIMULATOR (MS-DOS)	DM 948,–

Intec Electronic GmbH
Rheingrafenstraße 37, 6501 Wörrstadt
Telefon 0 67 32/50 29

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

Industrie- u. Meßkarten für PC, XT, AT

– 8-Bit-Analog-Digital-Wandlerkarte	129.–
– 12-AD/DA-Karte mit 16/1 Kanal, 0–9 V	248.–
– 35-MHz-Analog-Digital-Wandlerkarte, 8 Bit	548.–
– Relaiskarte mit 8 TTL-I/O und 8 Relais bestückt	249.–
– Optokoppplerkarte mit 16 Ein- und 8 Ausgängen	368.–
– Prototypenkarte mit 24 TTL-I/O	198.–
– 48-I/O-Karte mit 3×16-Bit-Timer und LED-Anzeige	248.–
– Schrittmotorenkarte für 2 Stepper	298.–
– EGA-/BAS-Wandlermodul, wandelt RGB in SW-Video	119.–
– Frequenzzählerkarte bis 1,3 GHz	298.–
– VD-8008, Videodigitalisierer, 1024 Pixel, 256 Graustufen	991.–
– BUS-EXTENDER-Karte für Servicearbeiten	98.–
– Slot-Erweiterung für PC/AT, 4 Steckplätze außerh.	256.–
– Netzwerkkarte EASY-Network mit LAN u. Software kompl.	290.–
– Modem-Karte, 300/1200 Baud, kompl. mit Software	339.–
– Akustikkoppler mit 300 Baud von Tandy	245.–
– Telefax-Karte kompl., Gr. 3 DBP, max. 9600 Baud	2900.–
– Drucker-Puffer-Karte mit 64 KB	264.–
– RS-232 ser. Schnittstelle, erweiterbar auf 2	89.–
– RS-422-Schnittstelle komplett	338.–
– IEEE-488-Schnittstelle komplett	780.–
– RS-232 auf RS-422-Wandlermodul, im Gehäuse	498.–
– 384-K-Multifunktionskarte, Uhr und sämtl. Schnittst.	189.–
– Multi-serielle Karte mit getrennten RS-232-Schnittst.	229.–
– Bar-Code-Pen mit Interface, viele Normen	791.–
– Handy-Scanner, Bildlesegerät, 62 mm breit	819.–
– Maus, MS-Mouse-kompatibel, u. Treiber	99.–
– Genius-Maus mit Treiber und Paintsoftware auf CGA	130.–
– GAME-I/O-Karte für PC mit 2 Eingängen	79.–
– Joystick (analog) für PC	36.–
– Eprommer LOW-COST bis 27256 Textoolsock. auf Karte	298.–
– Eprommer-Karte bis 27512 u. Textoolsock. extern	398.–
– Eprommer mit 4 Textoolsockeln extern	599.–
– PAL-Programmer mit 1 Textoolsockel	679.–
– 2-MB-RAM-Karte für XT, 0 K best., mit Treibersoftw.	329.–
– 386-AT-TOWER-Gehäuse u. 200-W-Netzteil	899.–
– CCD-1000, Industriekamera mit 500 × 500 Pixel, ab 3 LUX	1799.–
– 9-Zoll-Monitor, passend zur CCD-Kamera	248.–

Wir sind Hersteller von PC-Karten. Andere Karten auf Anfr. Bitte kostenl. INFO anfordern. Händleranfragen erwünscht. Versand erf. per NN. Auch EXPORT bei Vorscheck mögl.

COMPUTER-RING Heinrich Kolter Electronic
Steinstraße 22 · D-5042 Erftstadt
Tel. 0 22 35/7 67 07 · Fax 0 22 35/7 20 48 · Tx 8 882 134

IBM-XT+AT-kompatibel

80386-Hi-Tech

ICO 386-1 **4494.–**
80386 CPU, 20 MHz. 1,2 MB-Lauf-werk, deutsche Tastatur, serieller, paralleler Port.

ICO 386-40 MB **5595.–**
wie ICO 386-1 jedoch mit einer schnellen 38mS 40 Megabyte Festplatte und volle 2 MByte RAM. Höchste Rechenleistung für extreme Anforderung!

Maus 139.–
mechanisch MS-kompatibel an seriellen Port anzuschließen.

Barcodeleser 696.–
liest EAN, JAN, UPC, Codabar (NW7), 2 of 5 Interlave, Code 3 of9, Anschluß an Tastaturschnittstelle, dadurch keine Anpassungsprobleme

80286-Speed-Karte f. XT **359.–**
AT-Power auf dem XT! 6/8 MHz

VGA-Karte 696.–
alle VGA Modes bis 640x480 (bei 16 aus 256 Farben)
PS/2-Modell 30-kompotibel.

EGA-Karte (XT, AT) (640x350)	279.–
Super-EGA-Karte (640x480) Genoa kompatibel (XT, AT)	289.–
Super-EGA-Karte (800x600) Genoa kompatibel (XT, AT)	299.–
Seriell-Parallel-Karte (XT, AT)	99.–
22-MB Festplatte m. Controller u. Kabel f. XT	628.–
22-MB Festplatte ohne Controller	498.–
40-MB Festpl. (40 m/Sek.)	999.–
Monitor Grün (BAS-Anschluß für 12")	199.–
14 Zoll-Monitor (S/W 14")	299.–
EGA-Monitor (14")	798.–
Multisync-Monitor	1199.–

DIN-A3-Plotter PLX-88

Spitzentechnologie zum Sensationspreis

Made in Germany

Preis DM **1498,–**

- DIN-A3-Plotter
- 4 Farben m. autom. Stiftwechsel
- 0,025 mm Auflösung
- max. 200 mm/s Zeichengesch.
- HPGL-Kompatibel 28 Befehle
- Serienmäßig zwei Schnittst.
  Centronic u. RS 232 (V.24) ohne Aufpreis
- Ladbare Zeichensätze

Emis GmbH
Gesellschaft f. Electronic u. Microprozessorsysteme mbH,
Sebastianstraße 5, D-8480 Weiden, Telefon 0961/35053

ICO AT-1 **1799.–**
80286-CPU, 6/8/10 MHz, 640 K RAM, Color-Karte, 1,2 MB-Disk, deutsche Tastatur.

ICO AT-20 **2599.–**
Der Standard-AT: AT-1 m. 20 Megabyte Festpl.

ICO AT-30 **2699.–**
Wie AT-1 jedoch mit 30 MB-RLL-Platte.

12-MHz-Speed + 1MByte **292.–**
(statt der 640 KB) für obige ATs. Die 12 MHz vertragen sich durch neuartige Tastanpassungsschaltung. (DSC) auch mit Zusatzkarte.

AT-Konfigurationen

statt Color-, eine Monokarte	+10.-
statt Color-, eine EGA-Karte	+180.-
extra serieller + parallele Schnittstelle	+99.-

ICO 360 **888.–**
IBM XT kompatibler Rechner mit 8088 CPU 256 KB RAM, 360 KByte Laufwerk, Colorkarte, deutsche Tastatur, im stabilen Metallgehäuse im original-look.

ICO 20MB **1499.–**
Der XT-Profi. Wie ICO 360 jedoch mit einer 20 MegaByte Festplatte eingebaut.

Andere XT-Konfigurationen

Monochromkarte statt Colorkarte	+10.-
Extra Drucker-, Serielle-, Game-Port, Uhr	+70.-
8 MHz-Fastspeed (statt 4,77 MHz, original)	+40.-
Extra 2. Laufwerk	+249.-

Bondwell BW 8 **1599.–**
8088 CPU, 512 KB-RAM, LCD-Displ. mit 640x200 Punkte Grafik, 3,5" Disk, Uhr, serieller Port, Druckeranschluß u. Anschl. f. 2. Laufwerk, Akku-Betrieb. Inci. MS-DOS u. GW-BasicGewicht 4,5 kg.

5¼ Zoll Diskdrive zu BW 8 499.–

BW 8 S (Supertwist) **1699.–**
Mit Supertwist-Display, bsd. hoher Kontrast

7 Monate Garantie Versand erfolgt per NN. Händler: Bitte günstige Händlerpreisl. anfordern.

Klaus Jeschke
Hard-, Software
Adelheidstr. 2-20
6240 Königstein
☎ (06174) 3041

NEUE PC-/XT-/AT-PROZESSOR-EINSTECKKARTE

– 6 zusätzliche V.24-Schnittstellen für Ihren PC, COM1 bis COM4 – Beliebige Z80-ECB-Buskarten für Ihren PC – Dual-ported-RAM-Kommunikation

Wir entwickeln für Sie Hard- und Software. Unsere Spezialgebiete sind: BTX-DECODER, PC-HARD- und -SOFTWARE, STEUERUNGEN und SPEZIAL-COMPUTER: 68000, 8086, 80286, Z80

Ing.-Büro Kree/Panzarella
Ferdinandstraße 2
6380 Bad Homburg v. dH.,
Tel. 0 61 72/2 27 47

Hans-P. Stein SOFTWARE
Niederhöchststädter Str. 18,
6374 Steinbach/Ts.,
Tel. 0 61 71/8 51 94

BTX-DECODER, INHOUSE-ZENTRALE, PC-KARTEN, ECB-KARTEN, PC-BTX-SOFTWARE-DECODER, PC-BTX-EDITIER-SOFTWARE, CGA/EGA –> FBAS-VIDEO-WANDLER, KAMERA-SCANNER, NETZWERKE ISDN, NC-DATENSTATION, DFÜ-PLATINEN-LAYOUT SPS

Der Eprommer für IBM und -kompatible Apple //e, //gs,][+, und CPC 464/664/6128

✱ Programmiert alle gängigen EPROM- und EEPROM-Typen (z. B. 2716, 27C16, 2732, 2732A, 27C32, 2758, 2764, 2764A, 27C64, 27128, 27128A, 27C128, 27256, 27C256, 2508, 2516, 2532, 2564, X2804A, X2816A, X2864A ...) ✱ Menügesteuerte Software auf Diskette/Kassette ✱ 32 KByte frei für EPROM-Daten (Brennen des 27256 ohne Nachladen) ✱ Kein Umschalten, Stecken oder Löten nötig ✱ Programmierspannungen werden im Gerät erzeugt ✱ Verbindung zum Rechner über Flachbandkabel ✱ Rote und grüne LED zur Betriebsart-Anzeige ✱ Komplett mit 28pol. Textool-Sockel ✱ IBM- und CPC-Version mit Interface-Karte (durchgeführter Expansionsport bei CPC-Version) ✱

Preise für IBM: Fertiggerät **DM 399.50** ✱ Bausatz **DM 349.–**
für Apple: Fertiggerät **DM 269.50** ✱ Bausatz **DM 219.–**
für CPC 464/664: Fertiggerät **DM 289.50** ✱ Bausatz **DM 239.–**
für CPC 6128: Fertiggerät **DM 319.50** ✱ Bausatz **DM 269.–**
CPC-Software auf 3"-Diskette statt Kassette: + DM 15.–

DOBBERTIN
INDUSTRIE-ELEKTRONIK GmbH
Brahmsstraße 9, 6835 Brühl, Tel. (0 62 02) 7 14 17

Zeichnungen

einfach mit dem Computer erstellen!
E-CAD-Software CG87 der neuen Generation erstellt aus Ihren Angaben: Schaltungs-/Platinenvorlagen, Verbindungsliste, Stückliste, Bestückungs-, Bohrplan mit höchstauflösender Grafikkarte bis 1200 × 873 Pixel in 16 Farben, lauffähig auf IBM PC-XT/AT.
Leistungsmerkmale: Mehrlagentechnik, SMD-fähig, Symbol- u. Makrobibliotheken, Gummibandtechnik, Logiktest, Fangroutinen, Auflösung bis 1/1000 Zoll. Entflechtungshilfen: optimierte Schaltungs-, Layout- und Autorouter.
Ausgaben: Plotter, Photoplot, Laserdrucker, Matrixdrucker, Diskette, CNC-Steuerungen.

Programme zur Layout- und Schaltbilderstellung gibt es bereits für **DM 968.–**.

Photoplotservice v. CAD FILE schnell u. preiswert.

Grafikkarten

Höchstauflösend mit bis zu 1280 × 1024 auf dem Bildschirm sichtbaren Punkten und bis zu 256 Farben pro Punkt. Bildspeicher bis 2048 × 2048 Punkte, für IBM- o. ECB-Bus-Systeme inkl. Software **ab DM 990.–**.

Info anfordern bei:
Klaus Gorny · Computer Soft & Hardware
Bülowstraße 16b · 1000 Berlin 30 · Tel. (0 30) 2 16 91 29

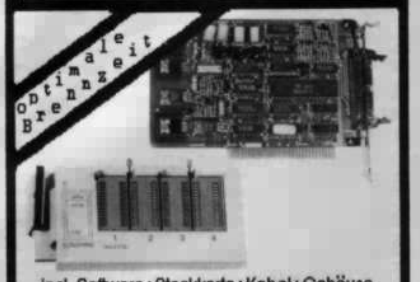
mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

IBM PC/AT und kompatible Zusatzkarten:

ADD-ON-CARDS

IC-Testgerät (74er & CMOS) DM 564.-
4-fach Eprom Brenner 16-512K DM 568.-



incl. Software+Steckkarte+Kabel+Gehäuse
*** + deutsche Beschreibung ***

Ingenieurbüro Lutger Ahlers, Mozartstr.23
8052 Moosburg, Tel. 08761/4245

Pocketmodem 1200 498,–
Nur 10 × 6 × 2,4 cm klein, für Portabels usw.
Modemkarte PC 1200 349,–
Kurze PC-Karte, V21/22, 300/1200 Bd vollduplex, RS 232 eingebaut (COM 1–4), Lautsprecher.
KS 1200, extern, sonst wie PC 1200 398,–
KS 2400, 2400/1200 Bd vollduplex 598,–
Multimodem, V21/22/23 598,–
Alle Modems Hayes-komp. und **ohne FTZ-Nr. Anschluß an das Postnetz nicht erlaubt.**
RAMboard 2 MB 349.–
EMS-/Intel-komp., ohne RAM, mit Software
AT-RAMboard mit 16-Bit Slot, sonst w. o. 449,–
AD/DA-Wandler 269,–
Auflösung 12 Bit, 16 AD-, 1 DA-Kanal
VIA 8255 179,–
I/O-Karte, 48 Leitungen, 16 LED, Timer/Counter
ARClink-Netzwerkkarte 790,–
Z. B. für Novell, Info über weitere ARClink-Produkte anfordern

bsb
Datentechnik GmbH
Hohenzollernring 74
5000 Köln 1
Telefon **02 21-13 14 41**

Reinhard Milde

(089) 7 69 46 31

Postf. 70 13 44, 8000 München 70

Floppy – Laufwerke

3,5" – 5,25" – 8"	500kB .. 2,0MB
z. B. SMD 280 – 3,5" – 1MB	179,00

Winchester – Laufwerke

3,5" – 5,25"	20MB .. 380MB
z. B. MS 3650 – 5,25" – 52MB uF	845,00
dto. mit XT – RLL – Controller (60MB)	995,00

Winchester – Controller

MFM, RLL, ESDI – XT, AT, kompatible
z. B. XT – Controller MFM, Kabelsatz 175,00

integrierte Schaltungen

6264lp – 15	10,95	27256 – 20	15,90
V20 – 8MHz	17,90	V30 – 8MHz	23,90

vollständige Liste anfordern!

C-COMPILER

MI-C für CP/M CP/M 86, MS DOS

MI-C vereint hohen Bedienungskomfort mit hervorragender Leistung

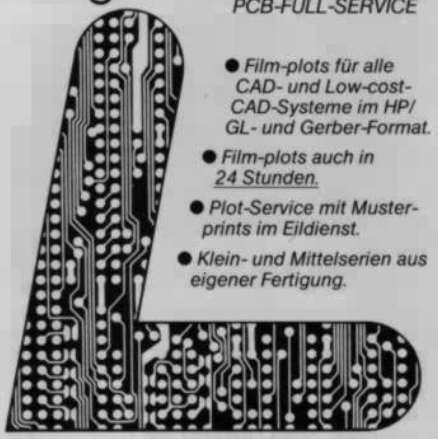
- Vollständige Version mit 13stelliger BCD-Arithmetik für Gleitkommazahlen
- Erzeugt kurze und schnelle ROMfähige Programme
- Ausgabe im Assemblercode des Zielprozessors
- Kompatibel zu MAC80/L80 (MASM/LINK) v. Microsoft
- Für 8086: 4 Speichermodelle/8087-Co-Prozessor
- Fehlerverfolgung mittels Trace möglich
- Umfangreiche Bibliothek (AMD9511-Paket erhältlich)
- UNIX-kompatibel
- Deutsche oder englische Version lieferbar 8"-/5,25"-/3,5"-/3"-Disk + dt. Handbuch

MI-C für CP/M	445.– DM
MI-C für CP/M 86 oder MS-DOS	575.– DM
MI-C Cross (Zielprozessor 8080/Z80)	745.– DM
MI-C Crossassembler + Linker (Ziel 8080/Z80)	645.– DM
MI-C Crosspaket (Ziel 8051)	1495.– DM
MI-C Crosscompiler/Assembler (Ziel 8096)	1495.– DM
MI-C Crossassembler (Ziel 68000/68010/68020)	795.– DM
MI-C Crossassembler (Ziel Z280)	795.– DM

Herbert Rose, Bogenstraße 32, 4390 Gladbeck, Telefon 0 20 43/2 49 12 und 4 35 97
Vertrieb in Österreich:
Dr. Willibald Kraml, Microcomputer-Software, Degengasse 27/16, A-1160 Wien

LAYTRONIC

PCB-FULL-SERVICE



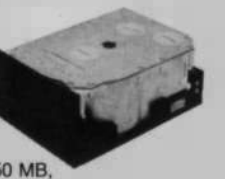
- *Film-plots für alle CAD- und Low-cost-CAD-Systeme im HP/GL- und Gerber-Format.*
- *Film-plots auch in 24 Stunden.*
- *Plot-Service mit Musterprints im Eildienst.*
- *Klein- und Mittelserien aus eigener Fertigung.*

LAYTRONIC GmbH · Industriestr. 7, 7434 Riederich
Telefon 07123/33635, 34223 · Telex 7245409, Fax 35105

Computer-Massenspeicher

Maxtor-Winchesterlaufwerke

mit Kapazitäten von 80 MB bis 765 MB in 5¼"-Ausführung. Zugriffszeiten von 29 ms bis 16 ms, mit Schnittstellen St 506/412; ESDI und SCSI zum Einsatz in:



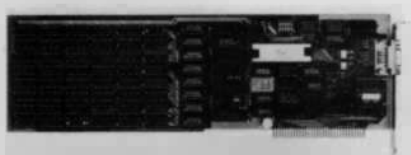
- PC-Mikrocomputer
- Multi-User-Systeme
- Netzwerke

Datensicherungen bis 150 MB, Speichererweiterungen 2–16 MB

- PS/2- und MS-DOS-Rechner
- XENIX/UNIX und Netzwerke

LANStor-Disk-Subsysteme von 80 MB bis 2932 MB voll Novell-286-kompatibel (V.2.0 und höher)

GBC Günther Benz Computer-Technik
Uracher Straße 45
Postfach 12 18
7433 Dettingen/Erms
Telefon (0 71 23) 7 10 81 / 74 86



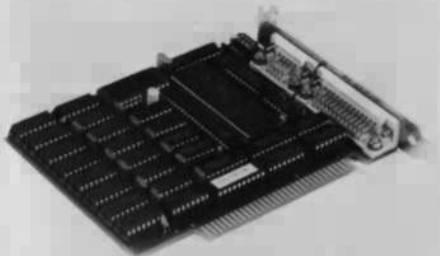
hochauflösende Farb-Grafik

für IBM/XT/AT (u. Komp.) mit SC-64

- NEC-7220, 8 MHz Highspeed-Version
- 1024 × 1024 Pixel, 8 Farben, 3 Mbit RAM
- 64 MHz oder 40 MHz Videotakt
- weitgehend kompatibel zu Artist 1
- Softwaretreiber in Pascal, C, usw.
- SC-64 Grafik-Board **DM 2950.–** + MwSt.

Mehr als 500mal bei bekannten CAD-Systemen u. anderen grafischen Anwendungen im Einsatz., Made by SC.

SC SPECIAL COMPUTER GMBH
LW. Hauptstr. 5 D-7630 Lahr
Tel. 07821/4990 Telex 754848.



PC-Subprozessor-Board

für IBM (kompatible) PC, XT, AT

- CPU HD64180, Takt 4,6 MHz
- 512 KByte RAM, max. 4 KByte EPROM
- Zwei serielle Schnittstellen, vollduplex
- Input 5 Bit parallel (Maus, Lesestift usw.)
- Dipschalter (3 Bit, von Software lesbar)

Viele Anwendungen, z. B. intellig. Druckerpuffer, Barcode, Datenerfassung usw.

SC SPECIAL COMPUTER GMBH
LW. Hauptstraße 5 D-7630 Lahr
Telefon 0 78 21/49 90 Telex 7 54 848

IBM-XT-Kompatible	ab **DM 998.–**
IBM-AT-Kompatible	ab **DM 1998.–**
Apple-Kompatible + Zubehör	ab **DM 99.–**

und natürlich alle Erweiterungen wie:

Genius-Mouse	ab **DM 118.–**
RAM-Karten für XT und AT bis 12 MHz, bestückt 2 MB	ab **DM 998.–**
Harddisk, 20 MB	ab **DM 498.–**
Floppy, 2× 40 Track	ab **DM 198.–**
TTL-Monitor, 14"-Flatscreen	ab **DM 398.–**
Matrix-Drucker	ab **DM 498.–**
Modem	ab **DM 398.–**
Btx-Karte IBM	ab **DM 598.–**

Und Service in der eigenen Werkstatt!

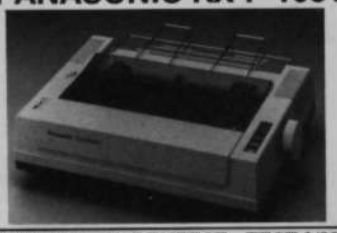
Fordern Sie für einzelne Produktgruppen unsere Preisliste an. Händleranfragen erwünscht.

ASC-ELEKTRONIK-VERTRIEBS-GMBH
HIRSCHGRABEN 9–11 · 5100 AACHEN
TEL. 02 41/2 52 26

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

PANASONIC KX-P 1081



STIFTUNG WARENTEST - TEST 2/88:

Im Test: 18 Nadeldrucker.
Testurteil: 7 gut*, 7 zufriedenstellend, 4 mangelhaft,
*Davon Panasonic KX-P 1081

gut

Hervorragende NLQ-Schrift in vielen Varianten, Schnellschrift mit 120 Z/Sek., umschaltbarer ASCII und IBM Zeichensatz, Centronics IF, eingebauter Traktor, Walze für Endlospapier und robuste zuverlässige Bauweise machen den KX-P 1081 zum preiswerten Drucker für viele Anwendungen.

KX-P 1081 - Unser Preis: . .nur 495,– DM

THEO WEBER ELEKTRONIK, 8700 WÜRZBURG
Eisenbahnstraße 22, Tel. 09 31-70 14 41

PANASONIC KX-P 1083

STIFTUNG WARENTEST - TEST 2/88:

Im Test: 18 Nadeldrucker.
Testurteil: 7 gut*, 7 zufriedenstellend, 4 mangelhaft,
*Davon Panasonic KX-1083

gut

Mit max. 240 Z/Sek., hervorragender NLQ Schrift, ASCII und IBM Zeichensatz, Centronics IF, eingebautem Traktor und Walze für Einzelblatt ein besonders schneller und zuverlässiger Universaldrucker. Natürlich auch Epson kompatibel.

KX-P 1083 - Unser Preis: . .nur 795,– DM
Vollautomatischer Einzug dazu: . **390,– DM**

THEO WEBER ELEKTRONIK, 8700 WÜRZBURG
Eisenbahnstraße 22, Tel. 09 31-70 14 41

AKTUELLE SONDERANGEBOTE:

Seagate Festplatte ST 225 - 21 MB	435,– DM
Seagate Festplatte ST 251 - 43 MB	895,– DM
Seagate Festplatte ST 4096 - 80 MB	1890,– DM
WD Controller 1002A-WX1 mit Kabel	145,– DM
Omti Controller 5520A mit Kabel	145,– DM
Archive Fastape FT20 Backupsystem	1295,– DM
SEKONIC Plotter SPL 410 - A3, 6 Farb.	1980,– DM
SEKONIC Plotter SPL 450 - A3, 8 Farb.	2250,– DM
SEKONIC Plotter SPL 430 - A3, 6 Farb.	2980,– DM
NEC Multisync II - neuestes Modell	1495,– DM
Getronics MC6710, 15,5-35 KHz. 14"	1250,– DM
NEC Multisync II mit Genoa Super Ega	1995,– DM
ATI EgaWonder Enhanced	450,– DM
Genoa Super Ega HiRes + GEM Graph	695,– DM
VICTOR 14" TTL Monitor bernstenfarben	350,– DM
Getronics FM 1400 - 14" Monitor, weiß,	360,– DM

Wir liefern Ihnen natürlich alle PANASONIC Drucker für jeden Leistungsbereich. Außerdem bieten wir Ihnen ein vielseitiges Lieferprogramm an Computern, Matrix- und Laserdruckern von OKI und NEC, Plottern und allem gängigen Zubehör. Gerne senden wir Ihnen auch unseren umfangreichen Katalog. Bitte anfordern!

THEO WEBER ELEKTRONIK, 8700 WÜRZBURG
Eisenbahnstraße 22, Tel. 09 31-70 14 41

Der Chef jubelt, die Kaffee-Händler fluchen!

Weg mit den überflüssigen Kaffeepausen!
Soft- und Hardware-Entwickler sparen jetzt wertvolle Zeit mit unserem Eprom-Simulator EPRAMMER V1.0. Direkt in der Schaltung simuliert EPRAMMER V1.0 bis zu 2 Eproms. Das zeitraubende Brennen und Löschen entfällt ab sofort!

Spezifikationen:
- direkte Unterstützung der Epromtypen 2764, 27128, 27256 und 27512, andere Typen (wie z.B. 27010) mit Spezialkabel.
- erhältlich in Versionen für 1 oder 2 Eproms (erweiterbar).
- minimale Stromaufnahme durch CMOS-Technik, deshalb keine zusätzliche Stromversorgung notwendig.
- superschneller Datentransfer durch Parallelübertragung, z.B. auf einem AT-kompatiblen Rechner mit 10 MHz-Takt nur 4 Sekunden für 64 Kilobyte.
- anschließbar an jeden Rechner mit Centronics-Schnittstelle, dokumentierte Treibersoftware gehört zum Lieferumfang.

Preise:
EPRAMMER V1.0 zur Simulation eines Eproms bis 27512: 798.- DM
EPRAMMER V1.0 zur Simulation von 2 Eproms bis 27512: 948.- DM
Informationsmaterial kostenlos.

Bezugsadresse:
Grewe Computertechnik GmbH, Richard-Wagner-Str. 73
D-4350 Recklinghausen, Telefon: (02361) 181354

moVe

Mohwinkel und Veiser GmbH

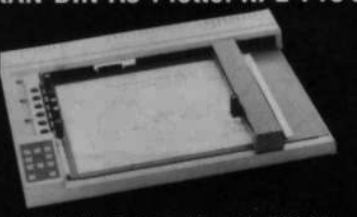
AUTOSCAN 20 DESKPROPAK
MICROVITEC-AUTOSCAN-20-Monitor
GENOA-SuperEGA-HiRes-Grafikkarte
HEWLETT-PACKARD-LaserJet Series II
LOGITEC-C7-Mouse
GEM Desktop – GEM Desktop Publisher – GEM Graph

inkl. Installation DM 14 998.– inkl. MwSt.
inkl. Schulung DM 15 998.– inkl. MwSt.

Bitte fordern Sie unsere detaillierten Unterlagen, auch zu anderen PC-kompatiblen Qualitätsprodukten von HP an.

5090 Leverkusen 1 · Berliner Str. 73
Postf. 25 01 66 · Tel. (02 14) 9 37 81-9 50 60

TAXAN-DIN-A3-Plotter KPL-710 S

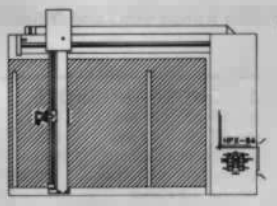


- Flachbett: alle Papiergrößen bis DIN A3
- Hochpräzise: Schrittweite 0,025 mm
- Schnell: 420 mm/Sekunde
- Kompatibel: mit HP-GL (AUTOCAD)
- Schnittstelle: Seriell RS232
- 2 Handbücher: User- + SERVICE-Manual
- Preis: 1899.– DM!

Und weiter Superpreise:
ATI EGA WONDER (ENHACED) 420.– DM
pass. 16"-Farbmonitor:
TAXAN-Super-Vision 840: 1420.– DM
(inkl. Service-Manual)

REGGE ELEKTRONIK · 28 BREMEN
Fesenfeld 57, Tel. 04 21/7 11 14

Plotter zu vernünftigen Preisen



DIN A3 Flachbettplotter HPX-84

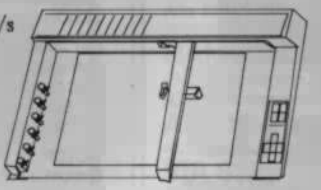
Technische Daten:
Plottflache 290 x 390mm
Wiederholgenauigkeit besser 0.1mm
Absolute Genauigkeit Abweich. max 1%
Befehlssatz HP-GL Kompatibel
Schnittstelle Centronics
Anzahl der Stifte 1
Auflosung 0.025mm

Preis DM 1698.-

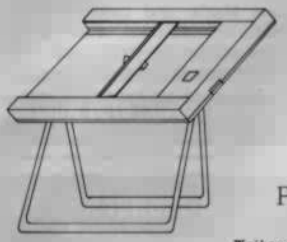
8-Farben DIN-A3-Flachbettplotter KPL-750

Zeichengeschwindigkeit 400mm/s
Zeichenflache 404 x 276mm
Wiederholgenauigkeit 0.2mm
Auflosung 0.025mm
Schnittstellen:
Centronic und RS-232c
56 HP-GL Befehle

Preis DM 2948.-



DIN-A1-Flachbettplotter HBS-101



Technische Daten:
Zeichenflache 810 x 620mm
Zeichengeschwindigkeit 120mm/s
Auflosung 0.025mm
Wiederholgenauigkeit 0.1mm
Schnittstelle Centronic
Option RS-232c
Befehlsatz HP-GL kompatibel

Preis DM 7498.-

Weitere Information bei

HBS-Grafiksysteme
Leonhard Habersetzer
Rigi Str.35 , 8123 Peißenberg , Tel.08803/2670

Rechner-Systeme
und Netzwerke mit

SPIDER

Deutsche Qualität
mit Höchstleistungen

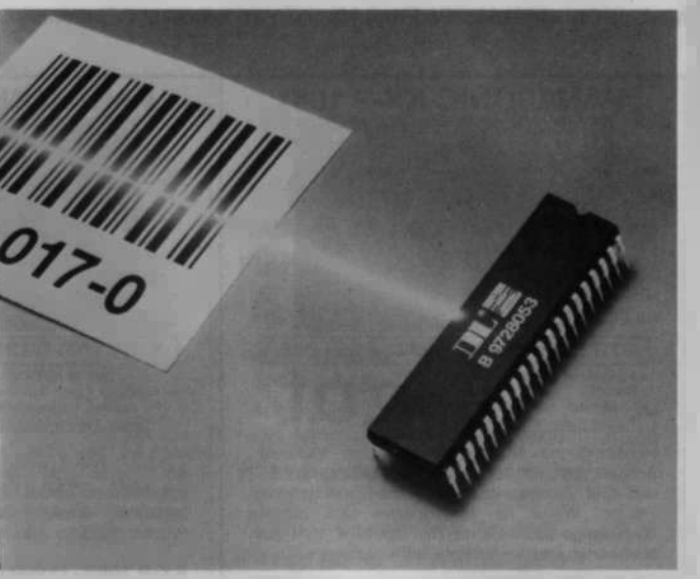
- Serie 16 und 32, kompatibel zum Industriestandard
- ideal für Dauerbetrieb in Desktop-Publishing, CAD, Netzwerk und Mehrplatzanwendungen
- bis 20 (24) MHz ohne Wartezyklen
- Hauptplatine von 1 MByte bis 8 MByte
- hoher Datendurchsatz durch Festplatten mit ESDI-Interface
- Ausbaufähigkeit und Leistungsreserven durch 350 Watt Netzteil
- anwenderfreundlich im fahrbaren Untertischgehäuse

SPIDER

Die einzige Alternative

Andreas Krischer, Noppiusstraße 19
5100 Aachen, Telefon (0241) 32896

KRISCHER
COMPUTERTECHNIK



Komfortable Strichcode-Erkennung mit dem CMOS-Baustein DL3

Einchip-Strichcode-Decoder

Mit einem minimalem Aufwand an externen Bauelementen lassen sich mit dem 40poligen CMOS-Baustein DL3 die TTL-Signale eines angeschlossenen Lesestiftes, eines CCD-Hand-Scanners oder einer Lesepistole in eine maschinenverständliche ASCII-Information umsetzen. Diese kann dann wahlweise seriell (Vollduplex) oder parallel ausgegeben werden. Der Baustein versteht alle gängigen Barcodes, mit einem umfangreichen Befehlssatz lassen sich alle Funktionen jederzeit neu festlegen; das umfaßt die Schnittstellen-Konfiguration, die Code-Auswahl, die Parität, das Reset-Verhalten usw.

Datalogic
Daimlerstr. 2
7311 Erkenbrechtsweiler
☎ 0 70 26/20 46

Kompakt trotz zwei Papierkassetten: Doppelseiten-Laserdrucker LBP-8II R von Canon

Laser druckt doppelseitig

Der Canon LBP-8II R bedruckt automatisch Vorder- und Rückseite des eingelegten Papiers – maximal 800 Seiten hintereinander, ohne daß die Papierkassetten (es gibt zwei davon) nachgefüllt werden müssen. Ansonsten besitzt der Drucker die in dieser Kategorie bekannten Eigenschaften: 300

Punkte/Zoll Auflösung und eine durchschnittliche Geschwindigkeit von maximal acht Seiten/Minute. Muß beidseitig gedruckt werden, verringert sich das Tempo um rund 10 %. Die standardmäßig verfügbaren Zeichensätze können durch zusteckbare Schriftkassetten erweitert werden.
In der Grundversion umfaßt der Pufferspeicher 512 KByte, er ist aber ausbaubar bis auf 2 MByte. Ein Vorteil des Doppelseiten-Druckes kann übrigens deutlich zu Buche schlagen: Die Halbierung des Papierverbrauches und damit auch günstigere Lagerungs- bzw. Archivkosten.

Canon Deutschland
Fraunhoferstr. 14
8033 Martinsried
☎ 0 89/8 57 00 10

Die 48-Kanal-Logikanalysator-Einsteckkarte gibt es mit vielen unterschiedlichen Prüflings-Koppeladaptern

Den PC als Logikanalysator einsetzen

In ihrer Grundversion erlaubt die Analysator-Einsteckkarte PA480 das Aufzeichnen von insgesamt 48 Digitalkanälen mit einer maximalen Abtastfrequenz von 25 MHz. Verschiedene Adapter gestatten die Kopplung des Logikanalysators sowohl an alle gängigen Prozessoren sowie an beliebige andere digitale Prüfobjekte.
Selbst schwer zu entdeckende komplexe Fehler können mit der bis zu 16 Ebenen tief strukturierbaren Triggerfunktion eingefangen werden. Ereigniszähler und Takt-Verzögerung lassen auch solche Ereignisse noch in den 4095 Worte fassenden Analysespeicher gelangen, die erst lange nach dem Triggerereignis auftreten.

Im Übersichtsdiagramm auf dem Computerbildschirm (EGA oder Hercules) lassen sich dann maximal 16 Kanäle simultan darstellen. Mit der Scroll-Funktion bzw. einem Auswahlmenü kann man sich genau die Signalkanäle auf den Bildschirm holen, deren Pegelzustände untersucht werden müssen. Vergrößerungen sind in fünf Stufen einstellbar, exakte Zeitmessungen kann man mit Cursor-Markierungen vornehmen.

Der Anwender hat natürlich auch die Möglichkeit der Ausgabe von Binär-, Oktal-, Dezimal-, Hexadezimal- oder ASCII-Tabellen. Für interessierte Anwender steht eine Demodiskette zur Verfügung.

gsh Software & Hardware
Postfach 60 01 55
8000 München 60
☎ 0 89/8 34 04 49

STAKOM
COMPUTER-ZUBEHÖR

Ihr Joker gegen hohe Preise

Diskettenboxen
DBS100 100 St. 5,25" mit Schloß 17,90
DBS050 50 St. 5,25" mit Schloß 14,80
DB3040 40 St. 3,5" mit Schloß 14,80

Computer-Werkzeug
11teilig! Mit Steckschlüssel, Greifer, Spezialpinzette, Schraubendreher, Abzieher und Einsetzer für IC.
CW005 Vinyl-Reißverschlußtasche 49,80

Reinigungs-Kits
Für ein- und zweiseitige Reinigung von Schreib-/Leseköpfen eines Diskettenlaufwerkes.
Lieferung: Diskette und Spezialreiniger
RK501 für 5,25" Laufwerke 24,80
RK301 für 3,5" Laufwerke 29,80
Allzweck-Reinigungs-Kit mit 2 5,25" Reinigungsdisketten, 1 Reinigerlösung, 10 Doppels. Reinigungsstäbchen, 5 Reinigungstücher, 5 Antistatik-Reiniger.
RK505 49,80
Mini-Staubsauger mit Batteriebetrieb
RK010 (Lieferung ohne Batterien) 24,80

Druckerständer
DS80132 mit verstellbarem Neigungswinkel 80-132 Zeichen-Drucker 29,80
DS80 mit Papierabl., 80 Z-Drucker 39,80

Bildschirmfilter
Vermindert Spiegelungen, schont die Augen und verbessert die Bild-Auflösung.
BF12C für 12" Color-Monitor 59,80
BF14C für 14" Color-Monitor 69,80
BF12M für 12" Monochrom-Monitor 49,80
BF14M für 14" Monochrom-Monitor 59,80

RS232/V24 Tester
TR01 Mini Tester, mit 7 LED's 39,80
TR05 für die Konfiguration einer RS232 Schnittstelle. Mit 24 DIP-Schaltern 2 x 24 Test-Pins, Kabel-Brücken 79,80
TR06 mit 8 LED, sonst wie TR05 98,00
TR02 mit DIL-Schaltern, Kabel-Brücken, LED, Klapp-Etui und deutscher Anleitung 228,00

Datenschalter/Verbindungskabel
DR252 D-25P, RS232, T 2-Weg 48,00
DR254 D-25P, RS232, T 4-Weg 65,00
DR25X D-25P, RS232, X-Schalter 78,00
DC362 36P, Centronics, T 2-Weg 51,80
DC364 36P, Centronics, T 4-Weg 68,00
DC36X 36P, Centronics, X-Schalter 88,00
IBM-Druckerkabel D-25P auf 36P
IDK362518 Centronics, 180 cm 16,80
D-25P, RS232, Verlängerungskabel
VKR25S18 Stecker-Stecker, 180 cm 14,80
VKR25018 Stecker-Buchse, 180 cm 14,80
VKR25B18 Buchse-Buchse, 180 cm 14,80
36P, Centronics, Verlängerung
VKC36S18 Stecker-Stecker, 180 cm 19,80
VKC36018 Stecker-Buchse, 180 cm 19,80
VKC36B18 Buchse-Buchse, 180 cm 19,80
D-25P, RS232, Modem-Kabel
MK25S18 Stecker-Stecker, 180 cm 19,80
MK25B18 Buchse-Buchse, 180 cm 19,80
MKPCAT18 IBM PC/AT, Modem-K., 180 cm 19,80

Druckerpuffer
Centronics-Buchsen, mit Netzteil
DP0641 1 Eing./1 Ausg., 64K RAM 198,00
DP0642 1 E./2 A. o. 2 E./1 A., 64K 298,00
DP2561 1 Eing./1 Ausg., 256K 328,00
DP2562 1 E./2 A. o. 2 E./1 A., 256K 398,00

Typumkehradapter
AR09S D-9P, Stecker-Stecker 8,80
AR09B D-9P, Buchse-Buchse 8,80
AR15S D-15P, Stecker-Stecker 9,80
AR15B D-15P, Buchse-Buchse 9,80
AR25S D-25P, Stecker-Stecker 14,80
AR25B D-25P, Buchse-Buchse 14,80
AC36S 36P, Centronics, St.-St. 19,80
AC36B 36P, Centronics, Bu.-Bu. 19,80
A09S25 IBM PC/AT, D-9P auf D-25P 19,80

Sonstiges Zubehör
Null Modems
NM25S D-25P, RS232, Stecker-Stecker 14,80
NM25O D-25P, RS232, Stecker-Buchse 14,80
NM25B D-25P, RS232, Buchse-Buchse 14,80
Jumper Box
JB25S D-25P, RS232, Stecker-Stecker 29,80
JB25O D-25P, RS232, Stecker-Buchse 29,80
JB25B D-25P, RS232, Buchse-Buchse 29,80
US01 Überspannungsschützer 29,80

Wir liefern nur hochwertige Qualitätsprodukte und bieten 1 Jahr Garantie auf jeden Artikel.
Keine Mindestbestellmenge!

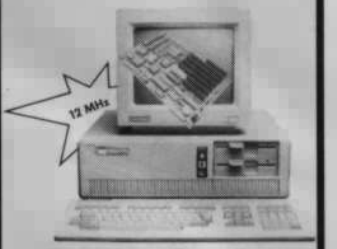
Rückgaberecht für alle Hardware-Artikel innerhalb 10 Tagen nach Erhalt der Lieferung.
Mit GELD-ZURÜCK-GARANTIE!

Versandkosten DM 9,—, Ausland DM 15,—.
Lieferung ab Lager per Nachnahme oder Vorkasse auf Psch.-Kto. Nr. 1936 47-701, PschA Stuttgart.
Firmen/Behörden Lieferung auf Rechnung möglich, Ausland nur Vorkasse, nur schriftliche Bestellungen an:

STAKOM Postfach 19 68 7030 Böblingen

MICROMINT
...damit alle IBM-Programme sicher laufen

MICROMINT
Power AT 286



2222.-
o. Monitor

6/10/12 MHZ in MICROMINT-Qualität
+ 512 K-1 MB RAM onboard
+ 101 Top-Profi-Tastatur
+ FDD 360/1,2
+ Hercules Card
+ serielle/parallele Schnittstelle
+ Text + Datei
+ dt. Handbuch
+ Monitor 14"/22 MHZ **289.–**
+ HD-Seagate/Contr. **799.–**
+ Genius GM6 Maus **99.–**

MICROMINT
● Händlerrabatte
● 8 Tage Rückgaberecht
● Prospekte anfordern

MICROMINT
bedeutet 40 Mio. Umsatz in 5 Jahren durch Vorsprung in Qualität und Preis.
MICROMINT bietet Spitzentechnik und HOCHLEISTUNGS-BIOS mit integriertem Setup, deutschem Zeichensatz und erhöhter Arbeitsgeschwindigkeit.

MICROMINT
zuverlässiger Service
Großlager
Tagestiefstpreisgarantie
8 Tage Rückgaberecht

MICROMINT Computer GmbH
Generalimporteur, Hochdahler Str. 151, 4006 Hochdahl
0 21 04 / 3 30 24

„Diskless Workstations", in den USA häufig zu finden, können jetzt auch bei uns in Netzwerke integriert werden

Schnelle Workstations fürs Netz

Die in den USA sehr populären „Diskless Workstations" des kalifornischen Herstellers Racore sind jetzt auch bei uns erhältlich, und zwar als 8-MHz-XT, als 10-MHz-AT und als 12,5-MHz-AT. Sie können mit verschiedenen IBM-Netzwerkadaptern (Token Ring) bei Übertragungsraten von 16 MBit/s geliefert werden. Für die Anwender von Netzwerken mit Arcnet- oder Ethernet-Topologie gibt es auch entsprechende LAN-Adapter. Den Bildschirm kann man sich übrigens als monochrome oder EGA-Version bestellen.

Molecular Computer
Hauptstr. 139–145
6236 Eschborn
☎ 0 61 96/43 50 38

Datenbank für Atari ST

Aus der englischen Software-Schmiede Kuma kommt jetzt das Datenbank-Programm K-Data in einer deutschen Version. K-Data ist sehr einfach anzuwenden und wird hauptsächlich mit der Maus bedient. Wichtige Kommandos können auch von der Tastatur gegeben werden. Die Datenmasken werden mit einem eigenen Programm erstellt. Maximal kann eine Datensatz-Maske 160 Zeilen und 174 Spalten enthalten. An Feldern stellt K-Data Textfelder, Zahlen-, Datum- und Zeit-Felder und „Klickfelder" zur Auswahl. Die Größe der Textfelder wird mit der Maus bestimmt, indem man die Felder auf die gewünschte Größe zieht. Dabei wird in der Kopfzeile der Umfang in Zeichen und Zeilen angegeben. Die Datenmaske kann mit freien Texten in beliebigen Größen und Arten versehen werden. Es läßt sich nur ein Schlüsselfeld definieren, das für die Sortierung und zur Verhinderung von doppelten Einträgen gebraucht wird. Suchen kann man in allen

AD
Computertechnik GmbH

Halmstraße 3
2820 Bremen 77
Telefon 04 21/37 13 59
Fax: 04 21/6 36 52 96

Ladenverkauf:
Münchener Straße 58

11 verschiedene ATs vom AT-Profi z. B.:
10 MHz/0 **waitstates** ab **1799.-**
12 MHz, CPU-Card **999.-**
15 MHz, DSC ab **2299.-**
12 MHz/0 waitstates ab **2299.-**

3 × 80386 von
16 MHz ab DM **4999.-**
bis **26,9** MHz ab DM **6699.-**

Viele Einzelkomponenten preiswert, z. B. 10 MHz/0 waitstate, AT-Mainboard DM 699.–

AD Turbo AT

Alle AT inkl. Hercules-komp. Karte + Colorgrafikemulation, AT-Gehäuse für 4 Einschübe, Slimline-Schlüsselschalter, Resettaster, Turboschalter bzw. Waitstateschalter, parallelem Printerport, 200-W-Netzteil, RT-Tastatur mit 12 Funktionstasten – separatem Cursorblock (deutsch oder ASCII), NEC-Diskdrive 1,2 MB (liest und schreibt 360K), 80287-Sockel, 512K (max. 1 MB on Board).

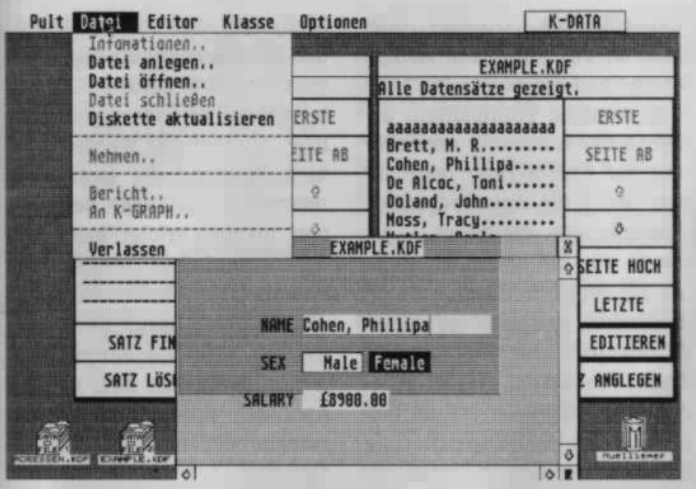
Genius Mouse
GM-6 DM **85.-**
GM-6 plus Maus kompl. DM 119.–
NEC P7 Drucker DM 1450.–

Handy Scanner **520.-**
Scanner + Texterkennung DM 1159.–
20 MByte **DM 479.–**

Turbo **XT** ab **929.-**

Einzelkomponenten, AddON, Netze, Drukker, Harddisk, AD/DA-Wandler, IEEE 488 und vieles mehr.

Sofort Katalog anfordern gegen DM 1.– in Briefmarken



Bis zu vier Dateien verwaltet K-Data auf dem Atari ST gleichzeitig

Feldern eines Datensatzes. Alle Funktionen werden im Dialog vom Anwender angefordert. Das Handbuch bezieht sich auf die Atari ST- und die PC-Version, die es auch gibt. Vergißt man, etwas in ein Feld einzutragen, oder hat man einen unerlaubten Text eingegeben, macht das Programm darauf aufmerksam, indem es das Feld einkreist und einen entsprechenden Text ausgibt. Bei der Anpassung der englischen Version hat es scheinbar Probleme gegeben, denn in einigen Meldetexten fehlen Buchstaben – oder es sind ein paar zuviel.

Bei einem ersten Test konnten wir feststellen, daß das Programm einfach zu bedienen ist und seine 130 DM wert ist. Es eignet sich für Einsteiger, die keine vollprogrammierbare Datenbank brauchen, sondern wirklich nur Daten mit dem Computer verwalten, sortierte Listen oder Etiketten erzeugen wollen. Selektierte Daten können auf Bildschirm, Diskette und Drucker ausgegeben werden. Mit den anderen Programmen von Kuma soll K-Data datenkompatibel sein. Für die Ansteuerung von beliebigen Druckern dürfte es keine Probleme geben, da K-Data über das beste Drucker-Konfigurierungs-Programm verfügt, das wir bislang für den Atari ST gesehen haben.

Knupe
Postfach 354
4600 Dortmund
☎ 0231/528033

mc-68000-Infos per Mailbox

Die DSM GmbH bietet allen mc-68000-Kunden sowie Interessenten ab sofort die Möglichkeit, aktuelle Informationen, Preise (viele wurden jetzt gesenkt), Liefermöglichkeiten und technische Ratschläge rund um die Uhr per Mailbox zu erhalten. Die dafür nötigen Kommunikations-Parameter: 300 Bd Vollduplex, acht Datenbit, ein Stopbit, keine Parität. Der Anruf bei der Mailbox (Tel. 0 89/59 33 26) muß im Originate-Modus erfolgen, Betriebsbereitschaft besteht rund um die Uhr.

DSM Digital Service
Landwehrstr. 37
8000 München 2
☎ 0 89/59 69 16

ELCO ELECTRONIC COMPONENTS GmbH

KOMPAKT-AT 10 MHZ

- CPU 80286-10 (80287 Option)
- umschaltb. 6/10 MHz, 0 Wait State
- aufrüstbar bis 16 MB (1MB on Board)
- 512 KB bestückt
- Uhr/Kalender batteriegepuffert
- Mono/Graphic/Printer-Karte (Herc.)
- Parallele Schnittstelle
- DIN Tastatur mit sep. Cursor/Z.-Block
- 1.2 MB Laufwerk
- Norton SI 11.5/Landmark 13
- stabiles Einschubgehäuse

ohne Monitor
DM 1.798,-

KOMPAKT-AT 13 MHz

- Konfiguration wie oben ● CPU 80286-12
- SI 15.3, Landmark 17,3

2198,-

TURBO-XT 4.77/10 MHz

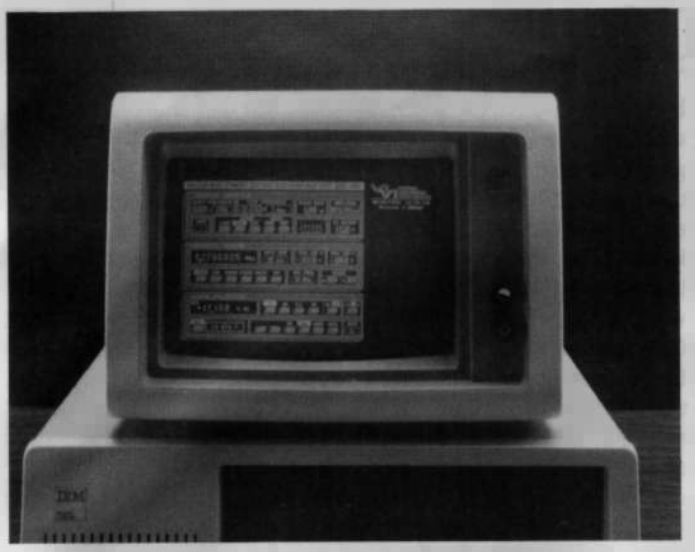
- voll PC/XT kompatibel
- CPU 8088-1/Sockel f. 8087
- 8 XT-Slots/256 KB RAM
- 1x360 KB FDD
- 150 Watt Schaltnetzteil
- Color Graphic Karte o. HGC/P
- Parallele Druckerschnittstelle
- Serielle Schnittstelle RS-232/Game Port
- Echtzeituhr/Kalender batteriegepuffert
- DIN Tastatur mit sep. Cursor/Z.-Block
- Stabiles Einschubgehäuse/ LED/Schlüssel

ohne Monitor
DM 998,-

Andere Rechnerkonfigurationen auf Anfrage!

- Typenraddrucker Silver Reed/20 Z/sek. ..... **DM 467,—**
- Epson LX-800/incl. Interface/Handbuch ... **DM 599,—**
- Mitsubishi 1471 ..... **DM 1498,—**
- 14" TTL Mon., entspiegelt, m. Standbasis ... **DM 278,—**

ELCO *Electronic Components GmbH*
D-6460 Gelnhausen-2 · Am Spielacker 18
☎ 06051/66088
Telex: 4184524 hzbm d · Fax: 06051/69205



Multimeter, Zähler und Funktionsgenerator als Einsteckkarte im PC – Frontplatten-Abbildungen auf dem Bildschirm: bei „Virtual Cat" ersetzt die Maus alle Einstellknöpfe

PC als Meßplatz

Basis des integrierten Meßsystems „CAT" (Computer Aided Testing) ist eine AT-Einsteckplatine mit der Hardware eines Digital-Multimeters, eines Zählers und eines Funktionsgenerators. Deren Anschlüsse sind nach außen geführt und können mit den Meß- oder Prüfobjekten verbunden werden. Da bei diesem „Meßplatz" natürlich keine Geräte-Einstellknöpfe, Displays oder Schalter zur Verfügung stehen, erscheinen auf dem Bildschirm drei virtuelle Frontplatten, auf denen der Anwender entweder mit der Cursor-Taste oder mit der Maus die entsprechenden Einstellungen vornimmt – genauso wie bei einem Einzelgerät auf dem Labortisch.

Besonders geeignet ist dieser PC-Meßplatz für Prüfaufgaben in der Produktion. Das mitgelieferte Softwarepaket erlaubt beispielsweise die Abspeicherung kompletter Geräte-Einstellungen, das Sichern von Ergebnissen einschließlich notwendiger Kommentare oder mit Datums- und Zeitmarken. Sämtliche Meßwerte sind so gespeichert, daß sie mit kommerziellen Programmen, z.B. Lotus 1-2-3, weiterverarbeitet oder analysiert bzw. ausgedruckt werden können. Während die Instrumenteneinstellung, wie beschrieben, manuell erfolgt, kann man den eigentlichen Meßvorgang und die Daten-Aufbereitung aus „Multitasking Basic" heraus aktivieren. Bereits in der mitgelieferten Software sind einige Applikationsprogramme für allgemein übliche Meßvorgänge enthalten.

Instrumatic Electronic Systems
Lohhamer Schlag 5 a
8032 Gräfelfing
☎ 0 89/85 80 20

Vielseitiges LAN

Nach dem bewährten CSMA/CD-Protokoll mit einer Transferrate von 1 MBit/s arbeitet das PC- bzw. PS/2-Netzwerk „D-Link". Als Übertragungsmedium dient eine zweiadrige Leitung, im Rechner selbst ist nur eine kurze Inter-

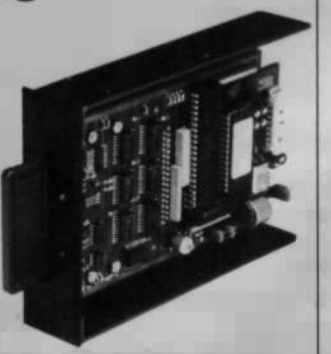
Von Omnitron ab Lager
Memory Card und Memory Card Read/Writer

Memory Card

- steckbarer SRAM-Speicher in CMOS-Technik
- Scheckkartenformat
- 8k bis 128kByte
- unempfindlich gegen statische Aufladungen
- Lange Betriebsdauer, bis zu 10 Jahren
- 256, 512kByte u. EEPROM MASKPROM, OTPROM-Versionen ab 2. Quartal 88 verfügbar
- Kartenschacht lieferbar

Panasonic

MPL AG



Memory Card Read/Writer

- preisgünstig
- serielles Interface RS 232
- hohe Transferrate
- geringe Leistungsaufnahme
- mechanische Abmessungen wie 3,5" Floppy Disk Drive
- auch als Tischgerät lieferbar
- Karte kann schreibgeschützt werden
- Daten sind durch Prüfcharakter geschützt
- komfortable Bedienerunterstützung
- extensive Fehlererkennung und Überprüfung
- Aufzeichnung von Datum und Zeit (optional)
- Speicherung von ASC II oder Binärdaten
- Daten werden in Records oder fixen Sektoren abgelegt
- integrierte Datenverwaltung
- kundenspezifische Protokolle möglich

Auch Ihre Batterieprobleme lösen wir für Sie mit
Panasonic

■ **Bleiakkumulatoren** ■ **Nickel-Cadmium-Batterien** ■ **Lithium-Batterien** ■ **Lithium-Batterien, wiederaufladbar**
Wir konfektionieren Batteriepacks nach Ihren Spezifikationen. Bitte fordern Sie Unterlagen an.

omnitron
ELEKTRONISCHE BAUELEMENTE

OMNITRON GRIESE GmbH, Kornweg 17, D-6229 Schlangenbad, Telefon 06129/4004, Telex 4 186 790 omnd, Fax 06129/8385

mc-MINIMARKT

Philips-Mikrocomp.-Entw.-System Typ PM4221 (PMDS-II) mit 68000, 256 KByte RAM, Winchester 5 MByte, UNIX-Betriebssyst., TRACE, DEBUG MAB f. NSC 800, Emulationsspeicher, Cross-Assembler Z80, EPROM-Programm. Promicron 2000, Centronix Druck. neuw. zu verk.
INTERFELS, 4444 Bad Bentheim,
☎ 0 59 22/7 26 66 Ⓖ

NDR-Computer mit Gehäuse: POW5V, SBC2, BUS, IOE, GDP, KEY, CAS MONI, 8K-Basic, VB 500.–. ☎ 0 52 02/7 17 61

2× Qume QT242 8"-Floppy DS/DD Slim-Line im Gehäuse, neuwertig, VB 2200 DM. V7 orig. ELTEC, 288-KByte-RAM-Karte, Bus-Plat., 19"-Rahmen, Tastatur, VB 1200 DM.
☎ 0 28 45/8 03 98, nach 19 Uhr

Apple 2e, 128 KByte, Duo-Disk, 80Z, SSC, SW-Monitor, alles orig., 1800 DM. ☎ 0 21 02/5 17 53

SONDERPREISLISTE für Monitorchassis + Netzteile.
☎ 0 51 46/86 81 Ⓖ

NDR-Computer, CPU 68008, 8 MHz, 660 KByte RAM, Floppy 3,5", 80 Spur, mit ser. u. Centr. u. 2× par. Schnittstelle, AD-Wandler, Joystick u. ca. 40 Disketten Software, Neupreis o. Software 3500 DM, für 1500.– DM zu verkaufen, event. Einzelverkauf 60 % unter BS. ☎ 0 23 30/7 34 43 (Herdecke)

4-FARBEN-WALZEN-PLOTTER DIN A4, Topzust., Parallel (Roland 800) usw. m. div. Software f. IBM-XT-Komp., f. DM 700.–.
☎ 0 77 74/76 36

IBS-Apple-Interf.: Z80-Karte AP22 mit 64 KByte u. 9-MHz-Takt, Zusatzkarte mit 256 KByte, Uhr und Schnittst. AP34, Grafik-Karte AP35, 512×512 Bildpunkte, 64 KByte, Grafikprozessor. Alle Karten mit Dok. u. Software. Zus. 800 DM. ☎ 0 61 09/2 25 14

Hewlett-Packard 21MX Computer mit Schaltplänen 500 DM, Floppy Shugart Model 400 40 DM, Printer Qume Model S3/45 120 DM, Burroughs Redactor Editing Typewriter 250 DM, Tastatur ITT 30 DM.
☎ 0 47 06/7 44

MC68000, 512 KByte, 1-MByte-Floppy, AD/DA-Wandler, Uhr, Busterm., 8-A-Netzteil, schönes Gehäuse, Monitor, voll funktionsfähig, viel Software, zum Preis von DM 1200.–. STAR-Drucker und EPROM-Brenner f. MC68000 ebenfalls abzugeben.
☎ 09 11/68 50 72

Wahnsinns-Preise
für Farbbänder,
Disketten, Diskboxen usw.
Fordern Sie unsere Liste an
BÜCOTECH
Postfach 1764-M, 5020 Frechen
☎ 0 22 34/1 56 92 Ⓖ

EPSON PX8 mit 120-KByte-RAM-DISK WORDSTAR u. a. 1750 DM. BROT EREP44 350 DM.
☎ 0 22 33/7 59 41

APPLE II+ Comp. + Monitor, 64 KByte + 2 LW (Siemens) + Z80 + 80 Z. + Time + Speech + SSC + I/O-6522, 68008, PROM Joy-S., Grappler + v. Softw. + Handb. VB 1900.–.
☎ 0 89/9 10 23 45, ab 19 Uhr

MC68020, MC68881 Chip-Pärchen 12 MHz DM 500.–.
☎ 0 56 73/71 65

Toshiba T3100, neu, 640 KByte, 10 MByte Festplatte, 720 KByte Floppy, VB DM 7800.–.
☎ 0 40/56 47 74, abends

Modem (Hayes), neu, 300–2400 Bd, 450 DM.
☎ 0 90 81/31 37, ab 20 Uhr

NDR-Klein-Computer:
Verkaufe folgende funktionsfähige Baugruppen:

3× Dyn. 256 **mit Speicher**	je 100.–
2× ROA 64 **mit Speicher**	je 100.–
1× CPU 68008	50.–
1× CPU Z 80	50.–
1× Flo 2	150.–
1× Bankboot	40.–
1× Cas	40.–

Dieter Kriesell, Badgasse 6, 8531 Lonnerstadt, ☎ 0 91 93/10 43

---

mc68000: 512 KByte, NT3, Floppy, Uhr, Sprache, 2 LW 3,5"/1 MByte, Gehäuse, MON1.4, CP/M68k, C, GEDIT, PHO..., Original-Handbücher, VB 1900 DM.
☎ 0 89/9 30 23 82

MC68000, 512 KByte, 2× 80-Spur-LW, Uhr, Sprache, Gehäuse, Tastatur, VB 1500.– DM.
☎ 0 69/38 81 44, ab 18 Uhr

NDR-Computerteile zu verkaufen, CPU 68k, ROA64, BUS2, GDP64k, Key, CAS, 2× IOE, Literatur, VB 400.–. ☎ 04 41/3 38 97

SAM68k, 256 KByte, Monitor, Tastatur, 2 Floppy, div. Software, DM 5000.–. ☎ 0 20 52/8 08 69

Verkaufe IBM-XT-Kompatiblen, 4,77/8 MHz, V20-CPU, V20-BIOS, 640 KByte RAM, Color- und Hercules-Grafikkarte, Multi-I/O-Karte, 1 LW 720/360 KByte schaltbar, 12"-Bas-Monitor, MS-DOS 3.20, für 1190 DM. XT-Mainboard, 4,77 MHz, 256 KByte RAM best., 150 DM. 512-KByte-Erweiterungskarte 70 DM.
☎ 06 81/89 31 73, ab 16.30 Uhr

NETWORK-SYSTEM unter CP/M, 4 Terminals, 54-MByte-Festpl., VB DM 9999.–. ☎ 0 75 51/8 00 00

15"-S/W-Monitor-Kit m. Netzteil ±50 kHz Zeile, 90.–. A/D-Wandler MC14433 10.–. ☎ 0 72 22/8 16 35

TOSHIBA T1200, 20 MByte HD, DM 6400.–.
ROHNER-EDV, Schelmenhofstr. 10, 8950 Kaufbeuren,
☎ 0 83 41/8 10 54, Tx. 5 41 290 Ⓖ

Unix/Xenix

Komplettlösung
Unsere Komplettlösung basiert auf dem UNIC-Tower (80286/80386) mit Multiprozessortechnik und ESDI-Festplatten und erreicht die Leistungsfähigkeit von Minisystemen.

Suchen Sie Erweiterungen für Ihre Systeme, bieten wir unter anderem folgende Produkte an:

Terminals:
VT 100, ANSI (ADI, Wyse, Ampex)

Festplatten:
390 MB ESDI/SCSI, 130 MB ST 506 RLL(MFM), 110 MB Interleave 1, 70 MB ST 506 MFM

Bandlaufwerke:
60 MB 1/4 Zoll, 120 MB 1/4 Zoll

Multiportkarten:
4/8 Ports, 6/8 Ports Intelligent (80161, 512 KB RAM)

Speichererweiterungen:
2 MB - 32 Bit, 4 MB - 32 Bit, 8 MB-32 Bit

Betriebssysteme:
SCO Xenix 286 Ver. 2.2, SCO Xenix 386 Ver. 2.2, Microport Unix V Rel.3

Applikationsprogramme:
Uniplex II Plus, Lyrix, Informix, Unify, Foxbase plus, Terminalemulation für DOS-Rechner etc.

aris edv-systeme

Wir beantworten gerne Ihre Fragen!

aris edv-syteme, haindlfingerstr.20, 8051 Palzing, tel.: 08167-8480 fax -1742

In aller Munde:
RIM electronic 88 die andere Art von Katalog

Völlig neu überarbeitete Ausgabe, über 1280 Seiten stark! Mit ca. 3000 Abb. und ca. 600 Schaltungen, Plänen, Skizzen, Anschl.-Bildern. Schutzgeb. 16,- DM. Versand: Vorkasse Inland 19,- DM (inkl. Porto), Postgirokonto München, 244822-802. Nachnahme Inland 22,20 DM (inkl. NN-Gebühr).
RADIO-RIM GmbH, Bayerstraße 25, 8000 München 2,
Postfach 202026, Telefon (089) 551720. Telex 529166 rarim d

Macintosh

Apple II

MS-DOS

HARDWARE SOFTWARE ZUBEHÖR BÜCHER
Spezialkatalog für ☐ Apple II, ☐ Macintosh, ☐ MS-DOS
Fordern Sie unter Angabe Ihres Rechnertyps den entsprechenden **Gratiskatalog** an!

pandasoft Dr.-Ing. Eden
Uhlandstr.195 D-1000 Berlin 12
Tel.: 030 / 31 04 24
Telex: 185 859

BITTE SCHICKEN SIE MIR IHREN GRATISKATALOG ZU!
Name: Adr.: Rech.Typ: MC

DAS TECHNISAT-KOMPLETTANGEBOT:

DM 1995,–

Multisat-Satelliten-TV-Empfangsanlage

(Aluminium-Parabolantenne 1,5 m + Polarmount + Polariser + LNC-SHF-Konverter** + Receiver TechniSat ST 2000 E** + 20 m Kabel)

TECHNISAT-PROGRAMMVIELFALT ZUM FAIREN PREIS!

PROSPEKTE SCHRIFTLICH ANFORDERN BEI:

Systeme von 0,99 bis 1,8 Meter erhältlich
** mit ZZF (FTZ)-Postzulassung

TechniSat

TechniSat GmbH · Postfach 1368
Industriegebiet · D-5568 Daun/Eifel
Telefon 0 65 92/45 23 oder 712-199 · Telex 4729936 srd d

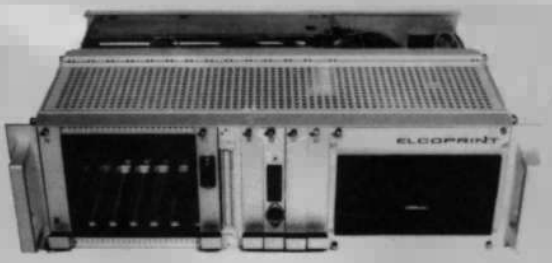
mc

Anzeigen-schluß-termine

Heft 6 vom	30.5.88
Anzeigenschluß	25.4.88
Heft 7 vom	27.6.88
Anzeigenschluß	24.5.88
Heft 8 vom	25.7.88
Anzeigenschluß	20.6.88

kompatibler

Industrie-PC



- **19" Einschubträger (3 HE) für den direkten Einbau in einen Schaltschrank oder eine Maschine**
- **CPU 80 86 oder 80 186, 8 (10) MHz Takt**
- **serielle und parallele Schnittstellen**
- **Floppy 1,2 MB, Festplatte 20 MB**
- **modular durch Baugruppen im Einfacheuropaformat**
- **7 bzw. 10 freie Steckplätze**
- **über 30 Erweiterungskarten verfügbar (Speicher, digital u. analog I/O, Bitbus, IEEE 488, etc.)**
- **Betriebssysteme: PC-DOS 3.2, CP/M 86, CCP/M 86, iRMX 86**

ELCOPRINT

GESELLSCHAFT FÜR VERBINDUNGS- UND DATENTECHNIK MBH

Member of AMBS*

Aidenbacher Str. 68
8358 Vilshofen
Telefon 0 85 41/30 82
Telex 57531
Telefax 0 85 41/30 81

Hannover-Messe, Industrie, H13/A24

*Systempartnerschaft zur Förderung von AMS-Systemen in industriellen Anwendungen

mc-MINIMARKT

BIETE AN HARDWARE

TA-alphatronic-P40 mit 8088-Coprozessor, 192 KByte RAM, 5-MByte- oder 12-MByte-Harddisk, 1-MByte-Floppy, Monitor, Tastatur, viel Software (CP/M80, CP/M86, MSDOS), VB 1100.–.
☎ 09 11/68 50 72

mc 68000, 1 MByte RAM, CP/M + Z80-Emulator, Tastatur, Monitor und Drucker, DM 1500.–.
☎ 0 81 42/27 27

Z80A-BULLET. ☎ 0 89/65 08 43

NEC-Monitore 12T311, 30-cm-Bildschirm, Fbe. silber, neuwertig, à 140.–. ☎ 07 11/31 52 91

MC68000-komp., 512 KByte, 40 MByte HD+, Cont., 2 LW, Gehäuse, Tast., Buster. CP/M68k, C, Forth, Editor, Preis VHS.
☎ 0 72 49/41 34, ab 17 Uhr

Tischcomputer HP 9845B, Basic, Assembler, 750 KByte Memory, HPIB, gegen Gebot.
☎ 0 70 32/2 36 75

NDR-Klein-Computer mit 68008-Prozessor, 384 KByte RAM, 2× 1-MByte-Floppy, in IBM-Gehäuse mit Tastatur und bernsteinf. Monitor sowie Betriebssystem CP/M 68K mit kompletter Original-Dokumentation (DRI, Franzis-Verlag) und div. Zubehör (Hard-/Software) für 1000 DM abzugeben.
☎ 05 21/88 59 96

Rechner für Z80-EMUF-Entwickl.: CP/M-2-EPC, 4× V.24, 1× Centr., 1× Par., 2× 800-KByte-Disk, Eprommer (eig. CPU), Term1, Video7, bernst. Monit., Siemens-Tast., NLQ-Drucker, SW + Doku. 1399.– DM.
☎ 0 25 94/16 38, nach 18 Uhr

Floppy-Disk Teac55FV, 720 KByte, DD, DS, 230.– DM; Tastatur Cherry G80-413 für XT 150.– DM; ECB-Bus-Platinen: Z80B + 64 KByte, Diskcontr., Grafik. NEC 7220. ☎ 02 28/66 34 94

mc68000, 2,5 MByte, 1 NEC-LW 1,2 MByte, 40-MByte-Harddisk, SIO-Karte, EPROM-Karte, 68020 & 68881, Uhr, Maus, Software: OS-9/CP/M68K, Compiler, Textverarbeitung usw.
☎ CH 01/8 56 05 56

25-MByte-Backup-Cartridge-Streamer (f. jed. Rechner!) 390.–
8"-Shugart-Floppy (neu) 390.–
Schaltnetzteil 5 V/15 A + 12 V/6 A + 12 V/1,7 A + –12 V/0,1 A 390.–
Eurocom 2 mit 76 KByte RAM VHS
Alle Geräte mit ausführlichen Schaltplänen!
MEYER, ☎ 0 43 07/67 17

Logic Analyzer (Gould) K105-D, NP 40 000 DM, 1 Jahr alt, für 25 000 DM. ☎ 0 30/8 81 26 24

Verkaufe: Hanheld Tandy Mod. 100 mit reichhaltigem Zubehör sehr preisgünstig.
☎ 02 01/51 10 50

MC68010 an Selbstabholer: 2,5 MByte, V.24, Busterm., Uhr, 2× LW, Epromdisk, stat. RAM-Karte, Z80-Karte, Monitor, 10-A-Netzteil, geändertes BIOS; Ramdisk ist resetfest, Uhr und stat. RAM wird vom BIOS initialisiert, wahlweises Booten von Epromdisk (bzw. stat. RAM) oder LW (laufwerkloser Betrieb); komplett DM 4500.–, evtl. auch Einzelabgabe.
☎ 0 30/4 13 18 29

IBM Portable PC, 256 KByte RAM, Floppy 2× 360 KByte/5¼", grafikfähiger 23-cm-Bildschirm, Farbgrafikkarte, Druckeradapter (parallel), 2 Jahre, Preis VHS.
☎ 0 42 46/3 76

Verkaufe EPSON PX 8 mit 120-KByte-RAM-Disk, Wordstar und diverse Software. Verhandlungsbasis DM 1000.–. ☎ 0 42 71/42 22

Gewerbl. Anzeigen sind mit Ⓖ gekennzeichnet.

Apple-II+-komp., Monitor, 64 KByte, Z80 (CP/M); 80 Zeichen, 1 TEAC-LW, Lit., Software, Handbü., DM 750.–. Christiani-Kurs Mikroprozessortechnik, 4 Lehrbriefe + Mikroprozessorsys., NP 696.–, DM 350.–. ☎ 0 23 59/23 10

LW 5,25", 1/1,6 MByte, 230.–, Floppy-Contr. (AT) 360 KByte/1,2 MByte, 100.–, Drucker 150.–.
☎ 08 71/7 16 65

Genie III mit Siemens-Tastatur an Selbstabholer, 1000 DM.
☎ 0 87 41/74 50

AT-komp. Rechner 6/12 MHz, 80286-12, Coprozessor 80287, 1 LW, 1 Festplatte, 40 MByte, 28 ms, 1 MByte Speicher, 1 parallele u. 2 serielle Schnittstellen, Gameport, Mouse, gr. Tastatur, EGA-Karte, 14-Zoll-EGA-Monitor, Preis 5200.–. Zuschr. unt. Nr. MC 099

CASIO PB700 kpl. mit 4-Farb-Plotter + Kassette + Netzgerät, wenig gebr., 480 DM.
☎ 07 11/88 29 24

Matrix-Drucker, farbfähig, FUJITSU DX2100 DM 740.–, DL2400 mit Centronics- u. Parallel-Schnittstelle DM 1760.–. W. FASTENRATH, ☎ 0 22 46/71 32

ADCOMP X 100 Plotter, 4farbig, neuwertig, VB: 1100 DM.
Olaf Altrock, 3036 Bomlitz 1,
☎ 0 51 61/4 90 39

Gebr. OLIVETTI M15 mit DOS + CH + US-TAST sFr. 1400.–. Kompl. NORTH STAR SYSTEM Z80, 512 KByte, 2 num. Proc. + SW mit Term. sFr. 600.–.
☎ CH 01/8 41 13 33, abends

Neuerscheinung

Lineare Optimierung für PCs

Die mathematische Beschreibung von Optimierungsproblemen und ihre Umsetzung in Pascal.
Von J. K. Böhm und D. A. Heinze.
196 S., geb.,
DM 78.–
ISBN 3-7723-8851-5

Dieses Buch zeigt, wie die Probleme der Optimierung auf einem PC ohne großen Kostenaufwand anhand von konkreten Programmen gelöst werden können.

Das Franzis-Fachbuch
Ein Qualitätsversprechen
Franzis-Verlag GmbH, Karlstraße 37–41, 8000 München 2

mc 4-88-4

Neuerscheinung

Multiplan 3.0 kompakt

Eine Einführung in das Tabellenkalkulations-Programm.
Von Dr. A. Pfeifer.
124 S., 30 Abb.,
kart. DM 16,80
ISBN 3-7723-2201-8
Dieses Buch verschafft dem Leser einen schnellen und verständlichen Überblick über die Möglichkeiten mit Multiplan zu arbeiten. Es zeigt ihm den Einstieg in die Benutzung des Planungs- und Tabellen-Kalkulationsprogramms Multiplan auf. Dabei werden die wichtigsten Eigenschaften des Programms erläutert und an zahlreichen Beispielen dargestellt.

Das Franzis-Fachbuch
Ein Qualitätsversprechen
Franzis-Verlag GmbH, Karlstraße 37–41, 8000 München 2

mc 4-88-19

Auftrag für Gelegenheitsanzeige in mc

Anschrift: Franzis-Verlag GmbH, Anzeigenabteilung, Karlstraße 41, 8000 München 2

Name und Adresse _____

_____ _____
Unterschrift/Ich zahle sofort nach Rechnungserhalt. Datum

☐ Preis für private Gelegenheitsanzeige je Druckzeile 4.– DM.
Chiffregebühr je Anzeige 8.– DM.

☐ Preis für gewerbliche Gelegenheitsanzeige je Druckzeile 14.82 DM (dürfen nicht unter Chiffre erscheinen). Bitte beachten Sie, daß **Preise in Anzeigen die MwSt.** enthalten müssen.

mc-programmbörse
☐ suche Software
☐ biete an Software

mc-minimarkt
☐ suche Hardware
☐ biete an Hardware

☐ Tausch ☐ Verschiedenes
☐ Kontakte ☐ Chiffre-Nummer
☐ vollständige Adresse
☐ Telefon

Ich wünsche folgenden Text in der nächstmöglichen Ausgabe zu veröffentlichen:

(Zutreffendes bitte ankreuzen)

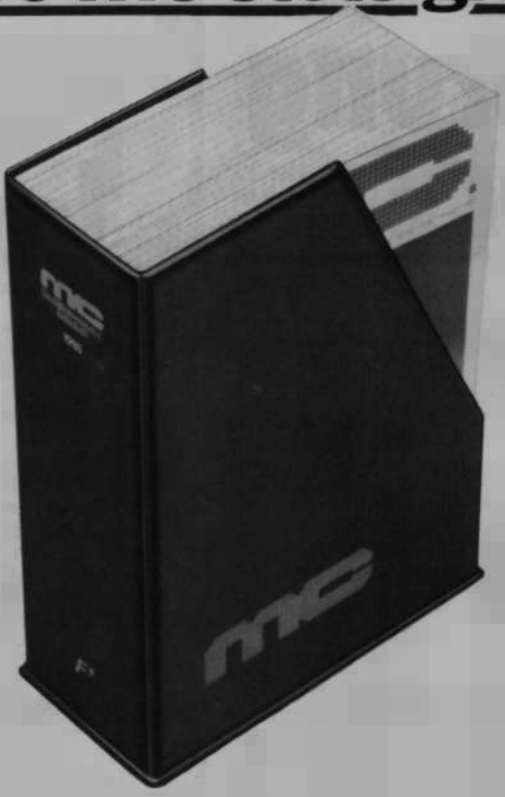
Bitte mit Schreibmaschine oder in Druckschrift in Groß- und Kleinbuchstaben ausfüllen.

Im praktischen Sammelordner bleibt Ihre mc stets griffbereit

mc-Sammelordner aus rotem Kunststoff für alle Hefte eines Jahrgangs. Das Selbstklebe-Etikett mit der Jahreszahl wird mitgeliefert.

Den Sammelordner erhalten Sie direkt vom Franzis-Verlag.

● Gegen Überweisung von DM 19,50 (DM 16,50 + DM 3,– Porto) auf unser Postgirokonto Nr. 813 75-809 Postgiroamt München (BLZ 700 100 80) mit Hinweis „mc-Sammelordner 1987" (bzw. welches Jahreszahl-Etikett Sie wünschen).



● Gegen Zusendung eines Schecks (DM 19,50).

● Gegen Rechnung.

Sofort nach Eingang Ihrer Bestellung senden wir Ihnen den Sammelordner zu.

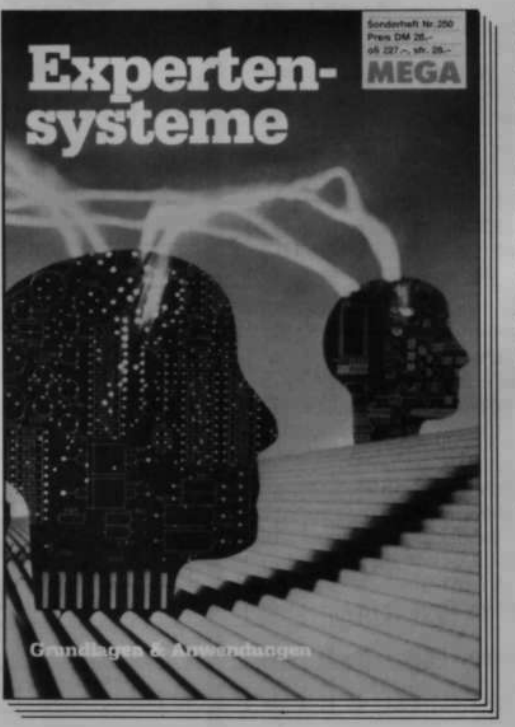
Franzis'
Franzis-Verlag
Kunden- und
Abonnement-Service
Karlstraße 37–41
8000 München 2
Tel. 0 89/51 17-2 79/-2 40

NEU

Expertensysteme

Grundlagen, Anwendungen, Märkte

Bei Expertensystemen, einem vielversprechenden Bereich der sogenannten „künstlichen Intelligenz", handelt es sich um wissensbasierte Informationssysteme, die den Erfahrungsschatz von Experten nutzen. Das von der Redaktion der Zeitschrift MEGA zusammengestellte Sonderheft „Expertensysteme" enthält eine Reihe von Beiträgen, die konkrete Anwendungen aus diesem Bereich schildern. So zeigt das Beispiel einer mittelständischen Maschinenbaufirma, bei der ein wissensbasiertes System das Herzstück einer rechnergestützten Fertigung bildet, daß nicht nur die „Großen" in Sachen künstliche Intelligenz mittlerweile recht fortschrittlich denken.
Neben solchen Anwendungsbeispielen bringt das MEGA-Sonderheft einführende Beiträge zur künstlichen Intelligenz und zu Expertensystemen.
Natürlich dürfen auch Informationen über diesen vielversprechenden Markt nicht fehlen:



Die MEGA-Redaktion hat in ihrem Special zwei Studien über den weltweiten Einsatz von Expertensystemen und das Geschäft mit LISP-Maschinen aufbereitet.

Bezugsmöglichkeiten:
Bei allen Bahnhofsbuchhandlungen, beim Elektronik-Fachhandel, in Mikrocomputershops, bei größeren Zeitschriftenverkaufsstellen, in Buchhandlungen oder direkt beim Franzis-Verlag mit Hinweis: Sonderheft „Expertensysteme"

● per Postkarte (Rechnung wird zugesandt)

● per Scheck (DM 28,– + DM 2,– Porto)

Franzis'
Franzis-Verlag, Karlstraße 37,
8000 München 2,
Tel. 089/51 17-240/-279
In der Schweiz: Thali AG, Fachliteratur und Software, Industriestraße 6,
CH-6285 Hitzkirch
In Österreich: Erb-Verlag,
Ges.m.b.H. & Co. KG, Mariahilfer Straße 71,
A-1061 Wien

Ein Sonderheft der **MEGA**

INSERENTENVERZEICHNIS

A.+L. Meier-Vogt 23
A.S.S.-Ware 138
Abor 34, 35
AD Computer 146
Ahlers 142
Aris 175
ASC 142
Atari 7

B & P 137
Bajic 139
Baron 17
Bartscher 139
Benz 142
Binder Datentechnik 101
Bockstaller 137
bsb Datentechnik 142

CE-TEC 61
Cesys 178
Chicony 160
Christiani 138
Computer-Ring Kolter 141
Computer Discount 178
Conex 34, 35

DaCom 33
Data Becker 11
Dawicontrol 173
De Ueno 156
Digitec 26
Dobbertin 141
DSM 8, 9
DVS 177

EcoSoft 139
Elco 147
Elcoprint 176
Elektronikladen 24, 25
Emis 141

Franzis 154, 155, 167, 179, 180
Fricke 139
Fujitsu 13

Gorny 141
Graf 21
Graphtec 149
Grewe 143
Grimberg 138
GWK 138

Haase & Menrad 140
Habersetzer 144
Hardline 168
HSK 177

IMTEC 139
Indutronic 3
Intec 140
Interconsult 177
Interest-Verlag 18
Interplanetary 156
Intra 169
iSystem 157

JAS 159
Jeschke 141

Kacer 171
Knauft 137
Koch CSM 139
Koga 151
Kree/Panzarella 141
Krischer 144
Kwem 31

Laytronic 142
Leadman 157
lehrer-pool 139
Linden, v. d. 137

Mathes 164
MCI 92, 93, 183
MEC 138
Meyer, E. W. 34, 35
Meyer Datentechnik 173
Micom 140
Microcomp 170
Micromint 145
Micro Parts 178
Microsoft 28, 29
Milde 142
Mitsubishi 15
Müller, Dr. Gert 140
Monterey 163
MoVe 143
MPK 138
MTP 178
MWB 174

Oberbeck 137
Omnitron 148

Pandasoft 175
PC Paradies 177
Plantron 2
Print Technik 140

Ranfft 140
Regge 143
Rheintec 152
Radio-RIM 175
Röntgen Software 138
Rose 142
RWL 150

Sanwel 162
Schukat 174
SC Special Computer 142
Shamrock 121
Siemens 42, 43
SOS 75
Stakom 145
Suchy 152

Technisat 176
Tetzlaff 138
Trans Computer 158
Trost 153
TSS-Schmitz 137

Ueding 174
Ultima 161

Vobis 184

Weber 143
Wefa 27
Weiss 138
Wiesemann & Theis 36
Wilke 137

Young Wisdom 158

Zenith 111

Die Mikrocomputer-Zeitschrift

Verlag: Franzis Verlag GmbH, Karlstraße 37–41,
8000 München 2
Postfach 37 01 20, 8000 München 37
Telefon (0 89) 51 17-1
Telex 5 22 301
Telefax (0 89) 51 17-3 79

Geschäftsführer:
Peter G. E. Mayer, Michael-Alexander Mayer

Verlagsleitung: Peter Habersetzer, Eugen Wintersberger

Anzeigenleitung: Dietger Kötter

Anzeigenverkaufsleitung: Hans-Joachim Hecht (-3 86)

Disposition: Ingrid Daschner (-2 97)

Anzeigenvertretung Inland:

Norddeutschland: Lita Lange, impulse medien service GmbH, Holtenklinker Str. 9, 2050 Hamburg 80, Tel. 0 40/7 24 20 37/38
Hessen: Günter Junne, Victor-Achard-Str. 30, 6380 Bad Homburg v. d. H., Tel. 0 61 72/3 33 94
Bayern: Elfie Rusch, Münchner Verlagsvertretung, Sperberstr. 23, 8000 München 82, Tel. 0 89/4 30 73 32
Nordrhein-Westfalen: Bandelow + Partner GmbH & Co. KG, Herr Werner Bandelow, Dellestr. 36, 4000 Düsseldorf 12, Tel. 02 11/20 14 64
Baden Württemberg: Ulrich G. Feiger, Honoldweg 27, 7000 Stuttgart 1, Tel. 07 11/63 27 18
Berlin: Rainer W. Stengel, Bischofsgrüner Weg 91, 1000 Berlin 46, Tel. 0 30/7 74 45 16

Anzeigenvertretung Ausland:

Belgien: ECI/United Media International S.A., Avenue de la folle chanson, 2 bte 7, 1050 Bruxelles, Tel. 02/6 47 31 90, Telex 63 950 eci um
Großbritannien: Martin Geerke, Friary Hall, Flat 3, Firary Road, South Ascot Berks. SL5 9HD, Tel.: 9 90/2 86 49 oder 2 76/6 28 28, Telex 8 58 328
Japan: International Media Rep. Ltd., 2–29, Toranomon 1-chome, Minato-ku, Tokyo 105, Tel. 5 02-06 56, Telex 22 761
USA: DemoNet Inc., 7310 Adams Street, Paramount, CA 90723, Tel. (2 13) 4 08-19 68
Frankreich: Agence Gustav Elm, 41, Avenue Montaigne, 75008 Paris, Tel. 01-47 23 32 67, Telex 2 90 260
Italien: Rancati advertising, Milano San Felice Torre 5, 20090 Segrate, Tel. 02-7 53 14 45, Telex 3 11 250 PP MI I
Schweiz: Exportwerbung AG, Kirchgasse 50, 8024 Zürich, Tel. 01-47 46 90, Telex 8 12 765

Anzeigen nach Preisliste Nr. 8, gültig ab 1. 10. 1987

Vertriebsleitung: Peter Habersetzer

Abonnement: Christa Fischer

Handelsverkauf: Dietlind Stauder

Erscheinungsweise: Die mc erscheint monatlich, jeweils montags am Monatsanfang bzw. am Ende des Vormonats; im 8. Jahrgang

Bezugspreise Inland: Einzelheft 7.– DM, Jahresabonnement 70.– DM; Vierteljahresabonnement 18.50 DM. Studenten, Auszubildende und Rentner erhalten das Jahresabonnement gegen Nachweis günstiger. Der Versand ist im Abonnementspreis eingeschlossen. In den Preisen ist die gesetzliche Mehrwertsteuer in Höhe von 7 % enthalten.

Verlagsvertretungen Ausland:

Belgien: Office International des Périodiques (O.I.P.), Avenue Marnix 30, B-1050 Brüssel – Einzelheft 182.– bfr, Jahresabonnement 1963.– bfr.
Dänemark: Harck + Gjellerups Booksellers Ltd., Fiolstraede 31–33, DK-1171 Kopenhagen K. – Jahresabonnement 316.– dkr.
Frankreich: Librairie Parisienne de la Radio, 43, rue de Dunkerque, F-75010 Paris.
Luxemburg: Messageries Paul Kraus, 5, rue de Hollerich, L-Luxembourg.
Niederlande: De Muiderkring BV, Electronics House, Postbus 313, 1380-AH Weesp – Einzelheft 8.85 hfl, Jahresabonnement 89.50 hfl.
Österreich: Erb-Verlag Ges.m.b.H. & Co. KG, Buch- und Zeitschriften-Vertrieb, Amerlingstr. 1, A-1061 Wien. – Einzelheft 60.– öS, Jahresabonnement 620.– öS.
Schweiz: Verlag Thali AG, CH-6285 Hitzkirch/Luzern – Einzelheft 7.– sfr, Jahresabonnement 67.20 sfr (je nach Kurs).

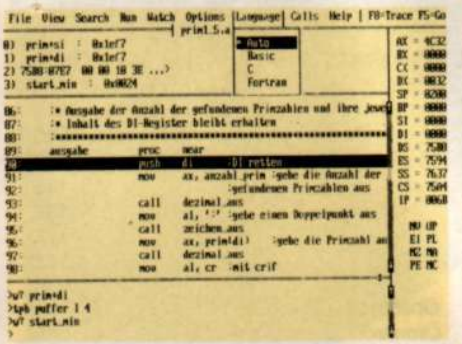
VORSCHAU

Peripherie-Netz

Messen, Steuern und Regeln mit dem PC – ein Thema für immer mehr Anwender. Doch was tun, wenn die verschiedenen Sensoren und Stellglieder räumlich weit verteilt sind? Da gibt es zwei Möglichkeiten: Dicke Kabelbäume verlegen oder einfach die nächste mc lesen. Dort wird nämlich Omega-Net vorgestellt: ein Low-Cost-Peripherie-Netz für PCs, mit dem sich solche Probleme preiswert lösen lassen. Zum Nachbauen und Nachdenken.

Aufbereitung von Bildern

Störungen können in digitalisierten Bildern durch mathematische Verfahren eliminiert werden. Fouriertransformationen eignen sich zum Beispiel zur Korrektur von Verwacklungen und Defokussierungen. Die Algorithmen werden im zweiten Teil des Artikels zur Aufbereitung von Bildern vorgestellt, zahlreiche Listings zeigen deren Umsetzung in Fortran-Programme.

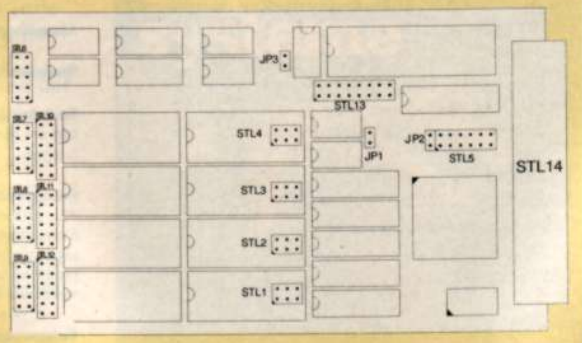


MASM 5.0 im Test

Microsoft bietet seit kurzem eine neue Version des Makro-Assemblers an, den MASM 5.0. Ist die Schnittstelle zu compilierten Programmen im Handbuch beschrieben? Ist MASM 5.0 kompatibel zu seinen Vorgängern? Lohnt sich der Umstieg? Erleichtern die vereinfachten Segment-Direktiven den Einstieg in die Programmierung in Assembler? mc beantwortet die Fragen der Anwender.

Transputer, Teil 2

Neben den Grundlagen geht es jetzt an die Schaltung: Timing und Adressierung werden besprochen. Dabei zeigt sich, wie leistungsfähig der T414 ist. Die mc-Transputerkarte kostet mit Cross Assembler 1750 DM. Sie wurde von einer kleinen innovativen Firma entwickelt – die Adresse gibt's bei der Redaktion.



Amiga programmieren

Für alle, die dem Betriebssystem Intuition mit C zu Leibe rücken wollen, gibt es im nächsten Heft den zweiten Teil unserer Serie. Und auch die Assembler-Programmierer läßt mc nicht im Stich: Diesmal verrät Spezialist Peter Wollschläger einen Trick, der viel Tipperei sparen kann – im Amiga-ROM steckt nämlich mehr, als es auf den ersten Blick scheint...

Aktuelle Sammeldisketten zu mc

MS-DOS-Sammeldisk 12
Diskette zum „PC-Sonderheft" (u. a. CLK.ASM, BAUM.ASM, TERM.BAS, DRUCKER.ASH, Hercules-Grafik, PC-Datentransfer, EGA-Grafik, Gameport, Disk-Speedy, Disk-Reparatur, Musik nebenbei). Preis: 25 DM.

MS-DOS-Sammeldisk 13
Diskette zum mc-Sonderheft „Turbo-Pascal im Einsatz" mit Pascal Quellcode.
Preis: 25 DM.

MS-DOS-Sammeldisk 14
Cross-Assembler und EPROM, Move-Utility für MOS-DOS und Zeitanzeige in Fenstertechnik aus mc 5/87; Wahrheitstafel für Logik-PROMs und schneller Draw-Line-Algorithmus aus mc 6/87; Hardcopy von CGA-Farbgrafiken und Flächen-Füll-Algorithmus aus mc 7/87; vier Bildschirmseiten in Turbo-Pascal, MS-DOS mit Komfort und Extended Memory (> 1 MByte) beim IBM-AT aus mc 8/87. Preis: 25 DM.

MS-DOS-Sammeldisk 15
256K-EGA mit variabler Bildgröße, Hercules-Grafiken drucken, Physik und Gehirn (KI) und Backup ganz bequem aus mc 9/87, Primzahlen-Sucher, pseudozufällige Binärfolgen, Utility für die Hercules-Karte und Leiterbahnen entflechten aus mc 10/87.
Preis: 25 DM.

MS-DOS-Sammeldisk 16
Differentieller Dateivergleich, dynamische Bildausgabe, PC minimiert Schaltfunktionen und Turbo-Residenz (aus mc 11/87). MS-DOS-Filemanager und Treiber-Implementation in MS-DOS (aus mc 12/87). Außerdem: Jahresinhalt mc 1987 mit Editor.
Preis: 25 DM.

MS-DOS-Sammeldisk 19
Turbo-Pascal ohne Absturz, Linien in C (aus mc 1/88); GALs programmieren (aus mc 1–2/1988); Festplatte parken (aus mc 1/88). Preis: 25 DM.

Für den ATARI-ST gibt es vier Disketten:

ST-Sammeldisk 1
EPROM-Programmer und Taschenrechner aus mc 9/87 (TOS-Version vom 27. 11. 86 oder später nötig). Preis: 25 DM.

ST-Sammeldisk 2
Grafikdemo und Minimon aus mc 5/87.
Preis: 25 DM.

ST-Sammeldisk 3
3-D-Darstellung von Funktionen aus mc 7/87, Assembler-Routinen in GFA-Basic und Setzen des Uhrenbausteins aus mc 9/87.
Preis: 25 DM.

ST-Sammeldisk 4
Physik und Gehirn aus mc 9/87, Splitspeed und Modula-2-Co-Routinen aus mc 11/87, diverse Public-Domain-Programme (Achtung: zweiseitig formatiert). Preis: 25 DM.

Fragen Sie nach den neuesten Disketten.
Bestellung ☎ 0 89/59 54 68

Ausgabe 5/88 erscheint am 25. April 1988

MCI

PERSONAL COMPUTER

Bitte neueste Prospekte anfordern!

Auf alle Geräte 12 Monate Garantie. Preise gültig ab 1.12.87. Lieferbedingungen auf Anfrage. MCI MICRO COMPUTER INSTRUMENTS GMBH eingetragen AG Bergisch Gladbach HRB 2575 · Herstellung und Vertrieb von Microcomputern.

MCI

Bensberger Straße 252 · 5060 Bergisch Gladbach 2
Tel. (02202) 1080
Fax: (02202) 31009 · Telex: 8873518

Clevere Partner mit Freundschaftspreisen.

Bisher gab es die VOBIS-Angebote nur in den eigenen Vertriebsstellen. Aber jetzt erhalten Sie einen ausgesuchten Teil unseres Lieferprogramms auch in zahlreichen Filialen der PORST AG. Genauso kompetent und preiswert wie bei uns: Mit uneingeschränkter VOBIS-Garantie.

VOBIS PORST

Hier sind die PORST-Adressen in postalischer Reihenfolge:

1000 BERLIN
Europa-Center
1000 BERLIN 12
Wilmersdorfer Str. 122
2000 HAMBURG 52
Osdorfer Landstr. 131, Elbe EKZ
2400 LÜBECK
Königstraße 75
3000 HANNOVER 1
Georgstraße 12
3400 GÖTTINGEN
Weender Straße 49
4150 KREFELD 1
Ostwall 152

4200 OBERHAUSEN 1
Marktstraße 85
4400 MÜNSTER
Prinzipalmarkt 4
4630 BOCHUM
Ruhrpark-Einkaufszentrum
4690 HERNE
Bahnhofstraße 43
5000 KÖLN 1
Schildergasse 100
5600 WUPPERTAL 1
(Elberfeld) Poststr. 1-3
6000 FRANKFURT 1
Roßmarkt 17

6200 WIESBADEN
Langgasse 1-3
6239 KRIFTEL
Beyerbachstraße 6
6680 NEUNKIRCHEN
Im Blieszentrum
6780 PIRMASENS
Hauptstraße 14-16
6800 MANNHEIM 1
F1, 4 Breite Str.
7000 STUTTGART 1
Schulstraße 20
7100 HEILBRONN
Kilianstraße 2

7500 KARLSRUHE 1
Kaiserstraße 185
7850 LÖRRACH
Turmstraße 12
7913 SENDEN
Berliner Straße 24
7980 RAVENSBURG
Bachstraße 42
8070 INGOLSTADT
Ludwigstraße 1
8200 ROSENHEIM
Bahnhofstraße 15
8400 REGENSBURG
Donau-Einkaufszentrum

8500 NÜRNBERG 1
Karolinenstr. 3
8540 SCHWABACH
Am Falbenholzweg 1
8600 BAMBERG
Promenade 6a
8630 COBURG
Spitalgasse 5
8720 SCHWEINFURT
Markt 12-18
8900 AUGSBURG
Maximilianstr. 21
8940 MEMMINGEN
Kalchstraße 1

Und hier die ersten Freundschaftspreise (**gültig ab 29.3.88**):

POCKET-COMPUTER

CASIO FX 790
alter VOBIS-Preis [illegible].- jetzt nur noch **169.-**
CASIO PB 1000 incl. 3.5" Floppy MD 100
statt ~~1020.-~~ (Summe der Einzelpreise)
komplett nur **899.-** Sie sparen 121.- DM!
HEWLETT PACKARD HP 28 CD **444.-**

HEIM-COMPUTER

COMMODORE C 64-II incl. Farbmonitor 1802 +
BTX-Modul II statt ~~1147.-~~ (Summe der Einzelpreise)
komplett nur **999.-** Sie sparen 148.- DM!

68000-COMPUTER

ATARI 520 STM **475.-**
ATARI Monitor SM 124 ..... **475.-**
ATARI 520 STM incl. Monitor SM 124
statt ~~950.-~~ (Summe der Einzelpreise)
komplett nur **899.-** Sie sparen 51.- DM!

68000-COMPUTER

CardCraft 3.5" Floppy 1 MB für ATARI **288.-**
AMIGA 500 incl. HIGHSCREEN KP 548 Stereo-
Farbmonitor statt ~~1746.-~~ (Summe der Einzelpreise)
komplett nur **1599.-** Sie sparen 147.- DM!

PERSONAL-COMPUTER

SCHNEIDER PC 1512 incl. 30-MB-Platte und
Controller statt ~~2265.-~~ (Summe der Einzelpreise)
komplett nur **2049.-** Sie sparen 216.- DM!

MONITORE

THORN EMI Ferguson, 25 Mhz, gelb **109.-**
HIGHSCREEN KP 548, Stereo **648.-**

DRUCKER

SEIKOSHA SP 180 VC **399.-**
NEC P6 ..... **1095.-**
NEC CP6 Color **1345.-**

VOBIS MICROCOMPUTER

kompetent und preiswert

VOBIS MICROCOMPUTER, POSTFACH 1778, 5100 AACHEN, TELEFON 0241/50 00 81, TELEX 832 389 vobis d

VOBIS MICROCOMPUTER immer in Ihrer Nähe:
1000 BERLIN 30 Kurfürstenstr. 101 030/213 94 80
2000 HAMBURG Krohnskamp 15 040/2 79 46 76
2300 KIEL Sophienblatt 74-78 0431/67 86 22
2800 BREMEN Violenstraße 37 0421/32 04 20
3000 HANNOVER Berliner Allee 87 0511/81 85 71
4000 DÜSSELDORF Wielandstraße 21 0211/35 99 64
4100 DUISBURG 1 Friedrich-Wilhelm-Str. 30 0203/2 78 63
4150 KREFELD Ostwall 92 02151/80 07 93
4300 ESSEN Hyssenallee 3 0201/23 11 74
4600 DORTMUND Hamburger Straße 110 0231/57 30 72
4800 BIELEFELD Herforder Straße 106 0521/6 38 78
5000 KÖLN Mathiasstraße 24-26 0221/24 86 42
5100 AACHEN Viktoriastraße 74 0241/54 31 00
5100 AACHEN Großkölnstraße 60 0241/2 44 94
(gemeinsam mit Foto PORST)
6000 FRANKFURT Frankenallee 207/209 069/73 40 49
6400 FULDA Mittelstraße 19/21 0661/7 82 66
(gemeinsam mit Foto PORST)
7000 STUTTGART Marienstraße 11-13 0711/60 63 36
7500 KARLSRUHE Kriegstr. 27/29 am BGH 0721/37 82 68
7750 KONSTANZ Kreuzlinger Straße 18 07531/1 55 60
8000 MÜNCHEN Aberlestraße 3 089/77 21 10
8500 NÜRNBERG Vordere Ledergasse 8 0911/23 29 95
8900 AUGSBURG Jakoberstraße 16 0821/152349

SE-DESIGN